INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.862
(12/97)

SERIES X: DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

OSI applications – Transaction processing

Open Systems Interconnection – Distributed transaction processing: Protocol specification

ITU-T Recommendation X.862

(Previously CCITT Recommendation)

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

PUBLIC DATA NETWORKS	
Services and facilities	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalling and switching	X.50–X.89
Network aspects	X.90–X.149
Maintenance	X.150–X.179
Administrative arrangements	X.180–X.199
OPEN SYSTEMS INTERCONNECTION	
Model and notation	X.200–X.209
Service definitions	X.210–X.219
Connection-mode protocol specifications	X.220–X.229
Connectionless-mode protocol specifications	X.230–X.239
PICS proformas	X.240–X.259
Protocol Identification	X.260–X.269
Security Protocols	X.270–X.279
Layer Managed Objects	X.280–X.289
Conformance testing	X.290–X.299
INTERWORKING BETWEEN NETWORKS	
General	X.300–X.349
Satellite data transmission systems	X.350–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	
Networking	X.600–X.629
Efficiency	X.630–X.639
Quality of service	X.640–X.649
Naming, Addressing and Registration	X.650–X.679
Abstract Syntax Notation One (ASN.1)	X.680–X.699
OSI MANAGEMENT	
Systems Management framework and architecture	X.700–X.709
Management Communication Service and Protocol	X.710–X.719
Structure of Management Information	X.720–X.729
Management functions and ODMA functions	X.730–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	
Commitment, Concurrency and Recovery	X.850–X.859
Transaction processing	**X.860–X.879**
Remote operations	X.880–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999

For further details, please refer to ITU-T List of Recommendations.

ITU-T RECOMMENDATION X.862

OPEN SYSTEMS INTERCONNECTION – DISTRIBUTED TRANSACTION PROCESSING: PROTOCOL SPECIFICATION

Summary

This Recommendation describes the application layer protocol for OSI distributed transaction processing. The protocol provides a means to group a set of actions called a <transaction>. It also provides a framework for coordination of a transaction across multiple transaction processing resources in separate open systems.

Source

ITU-T Recommendation X.862 was prepared by ITU-T Study Group 7 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 12th of December 1997.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, the ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Page

Page

Introduction

This Recommendation is one of a set of standards produced to facilitate the interconnection of computer systems. It is related to other International Standards in the set as defined by the Reference Model for Open Systems Interconnection (see ITU-T Rec. X.200 | ISO/IEC 7498-1). The Reference Model subdivides the area of standardization for. interconnection into a series of layers of specification, each of manageable size.

The aim of Open Systems Interconnection (OSI) is to allow, with a minimum of technical agreement outside the interconnection standards, the interconnection of computer systems:

a) from different manufacturers;

b) under different management;

c) of different levels of complexity; and

d) of different technologies.

The ITU-T Recommendations X.860-series | ISO/IEC 10026 defines an OSI TP Model, an OSI TP Service and specifies an OSI TP Protocol available within the Application Layer of the OSI Reference Model.

The OSI TP Service is an Application Layer Service. It is concerned with identifiable information which can be related as distributed transactions, which involve two or more Open Systems.

This Recomendation provides sufficient facilities to support transaction processing, and establishes a framework for coordination across multiple TP resources in separate open systems.

This Recomendation does not specify the interface to local resources, nor does it specify an application programming interface within the local system.

Recommendation X.862

OPEN SYSTEMS INTERCONNECTION – DISTRIBUTED TRANSACTION PROCESSING: PROTOCOL SPECIFICATION

(revised in 1997)

1 Scope

This Recommendation provides:

a) a statement (see clauses 6 to 11) of the nature of the automaton giving the necessary behaviour of each of the participating entities which are providing the OSI TP Service, covering:

   1) the actions to be taken on receiving request and response primitives issued by a TP Service user invocation;

   2) the actions to be taken on receiving indication and confirm primitives issued by the presentation service-provider;

   3) the actions to be taken as a result of certain events within the local system;

   4) the actions to be taken as a result of interactions with other ASEs;

b) the definition (see clause 12) of the abstract syntax required to convey the TP protocol control information;

c) the conformance requirements to be met by implementations of this protocol (see clause 13).

The scope of this Recommendation is limited to the interconnection of systems; it does not specify or restrict the implementation of possible interfaces within a computer system.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation X.200 (1994) | ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*.
- ITU-T Recommendation X.207 (1993) | ISO/IEC 9545:1994, *Information technology – Open Systems Interconnection – Application layer structure*.
- ITU-T Recommendation X.210 (1993) | ISO/IEC 10731:1994, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services*.
- ITU-T Recommendation X.215 (1995) | ISO/IEC 8326:1996, *Information technology – Open Systems Interconnection – Session service definition*.
- ITU-T Recommendation X.216 (1994) | ISO/IEC 8822:1994, *Information technology – Open Systems Interconnection – Presentation service definition*.
- ITU-T Recommendation X.217 (1995) | ISO/IEC 8649:1996, *Information technology – Open Systems Interconnection – Service definition for the association control service element*.
- ITU-T Recommendation X.225 (1995) | ISO/IEC 8327-1:1996, *Information technology – Open Systems Interconnection – Connection-oriented session protocol: Protocol specification*.
- ITU-T Recommendation X.226 (1994) | ISO/IEC 8823-1:1994, *Information technology – Open Systems Interconnection – Connection-oriented presentation protocol: Protocol specification*.

- ITU-T Recommendation X.227 (1995) | ISO/IEC 8650-1:1996, *Information technology – Open Systems Interconnection – Connection-oriented protocol for the association control service element: Protocol specification*.

.– ITU-T Recommendation X.520 (1997) | ISO/IEC 9594-6:[1], *Information technology – Open Systems Interconnection – The Directory: Selected attribute types*.

- ITU-T Recommendation X.650 (1996) | ISO/IEC 7498-3:1997, *Information technology – Open Systems Interconnection – Basic Reference Model: Naming and addressing*.
- ITU-T Recommendation X.680 (1997) | ISO/IEC 8824-1:[1], *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*.
- ITU-T Recommendation X.690 (1997) | ISO/IEC 8825-1:1998, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*.
- ITU-T Recommendation X.851 (1997) | ISO/IEC 9804:1998, *Information technology – Open Systems Interconnection – Service definition for the Commitment, Concurrency and Recovery service element*.
- ITU-T Recommendation X.852 (1997) | ISO/IEC 9805-1:1998, *Information technology – Open Systems Interconnection – Protocol for the commitment, concurrency and recovery service element: Protocol specification*.
- ITU-T Recommendation X.880 (1994) | ISO/IEC 13712-1:1995, *Information technology – Remote Operations: Concepts, model and notation*.

2.2 Paired Recommendations | International Standards equivalent in technical content

- ITU-T Recommendation X.290 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – General concepts*.

  ISO/IEC 9646-1:1994, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 1: General concepts*.
- CCITT Recommendation X.292 (1992), *OSI conformance testing methodology and framework for protocol Recommendations for CCITT applications – The Tree and Tabular Combined Notation (TTCN)*.

  ISO/IEC 9646-3:1992, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 3: The Tree and Tabular Combined Notation (TTCN)*.
- ITU-T Recommendation X.296 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Implementation conformance statements*.

  ISO/IEC 9646-7:1995, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 7: Implementation conformance statements*.
- ITU-T Recommendation X.860 (1997), *Open Systems Interconnection – Distributed transaction processing: Model*.

  ISO/IEC 10026-1:1998, *Information technology – Open Systems Interconnection – Distributed Transaction Processing – Part 1: OSI TP Model*.
- ITU-T Recommendation X.861 (1997), *Open Systems Interconnection – Distributed transaction processing: Service definition*.

  ISO/IEC 10026-2:1996, *Information technology – Open Systems Interconnection – Distributed Transaction Processing – Part 2: OSI TP Service*.

3 Definitions

For the purposes of this Recommendation, the definitions given in ITU-T Rec. X.860 | ISO/IEC 10026-1 (TP Model) and ITU-T Rec. X.861 | ISO/IEC 10026-2 (TP Service), in addition to those given in 7.3, apply.

Definitions of terms specific to the OSI TP protocol specification are contained in 7.3.

[1] To be published.

4 Abbreviations

Abbreviations used in the OSI TP protocol specifications are defined in ITU-T Rec. X.860 | ISO/IEC 10026-1 (OSI TP model), except for the following, which are used in some tables:

cnf	confirm primitive
ind	indication primitive
req	request primitive
rsp	response primitive

and for the following, which are used as prefixes to auxiliary facilities services:

AF	Auxiliary Facility[COMP5]
CAF	Channel Auxiliary Facility[COMP6]
SAF	SACF Auxiliary Facility[COMP7]

5 Conventions

ITU-T Rec. X.861 | ISO/IEC 10026-2 defines services for Distributed Transaction Processing guided by the descriptive conventions defined in ITU-T Rec. X.210 | ISO/IEC 10731.

However, the terms "request" and "indication" are sometimes used in the following ways:

a) a single request may result in multiple indications (an example is that a single TP-COMMIT request can result in TP-PREPARE indications to each direct subordinate TPSUI);

b) several requests may result in a single indication (an example is that a single TP-COMMIT-COMPLETE indication may be issued to a superior TPSUI only after TP-DONE requests have been issued by the TPSUI and all subordinate TPSUIs in the transaction tree);

c) the convention that a request primitive results in an indication primitive of the same name is not always followed (for example, a TP-COMMIT request will cause a TP-PREPARE indication to be issued).

For a given primitive or APDU, the presence of each parameter or field is described by one of the following values:

blank	Not applicable
M	Presence is mandatory
U	Presence is a user option
O	Presence is a provider option
C	Presence is conditional

In addition, the notation (=) indicates that a parameter or field value is semantically equal to the value of the parameter or field of the preceding primitive or APDU in the table. This notation is in some instances combined with another value above, e.g. "(=)/M", and signifies that in some cases the primitive follows as a result of a preceding primitive or APDU [that is, "(=)" applies] and in other cases (when "M" applies), either:

i) there is no preceding primitive or APDU; or

ii) the value from the preceding primitive or APDU can be changed.

6 Model of the PM

6.1 Overview

This clause provides an overview of those aspects of the TPPM which are specific to this Recommendation. These include association usage and management, the details of dialogue establishment and channel management, the use of the Session synchronize-minor token, concatenation, and embedding.

6.1.1 Principles of association usage

An association is used by a TPPM to support either a:

- TP dialogue; or
- TP Channel.

An association may be established at any time, according to a local decision. The setting up of an association may be done in parallel with the actions of the PM. An association that has been established and is not currently being used is considered to be in a pool of free associations.

On receipt of a dialogue request, the PM needs to be assigned an association to support this dialogue. Any association that is assigned must have attributes compatible with the dialogue it is to be used for, as described in 8.5.2 (dialogues) and 8.5.3 (channels).

An association may be assigned to the PM from the pool of free associations, or attempts may be made to establish a new association for use with this dialogue or channel. If, as a local decision, it is decided that a compatible association cannot be assigned, the begin dialogue request will be rejected.

Associations may be released at any time they are not in use by the TPPM. The point at which an association becomes unused, and therefore may be released, is defined in the SACF procedures in clause 10.

On the establishment of an association, one AEI is assigned to be the "contention-winner" and the other as the "contention-loser". The assignment of contention-winner and contention-loser remains for the duration of the association. An AEI may be the contention-winner on some associations and contention-loser on others.

The direction from the contention-winner to the contention-loser is the preferred direction of dialogue establishment because the contention-winner has the right of use of the association. The contention-winner may grant to the contention-loser the use of the association for the purpose of establishment of a dialogue, if it is not using or has not reserved this association. The contention-winner may also deny the use of the association by the contention-loser for the purpose of dialogue establishment.

The contention-loser may formally request the rights of the contention-winner temporarily in order to attempt establishment of a single dialogue. This is done using the bid mechanism. The use of the bid mechanism is declared to be either optional or mandatory at association establishment time. This declaration does not change for the life of the association.

6.1.2 Functional unit capabilities and selection

At association establishment, the TPPMs negotiate which of the TP functional units are supported by both of them. The extensibility rules for the TP-INITIALIZE-RI and TP-INITIALIZE-RC APDUs are such that this negotiation can succeed, and the association be established, even if one TPPM proposes the use of a functional unit that is not recognized by the other. Using this shared knowledge of joint capabilities, subsequent dialogue and channel establishment attempts are constrained to propose only functional units that are known to be supported by both TPPMs. There is no negotiation of functional units at dialogue establishment other than the ability for the responder to reject the attempt and suggest different functional units that would be acceptable. Dialogue establishment may fail because the TPSU addressed is unable to support the functional units, but the TPPM is assumed to at least recognize the functional units proposed on at dialogue establishment.

In addition to the functional units that are visible in the TP service, and described in ITU-T Rec. X.861 | ISO/IEC 10026-2, the following functional units affect only the TP protocol:

a) **Solicit Dialogue**: The Solicit Dialogue functional unit allows an AEI to request the peer to establish a dialogue on an existing association to a TPSU at the first AEI. If successful, the dialogue will be established with the TPSU at the AEI that solicited the dialogue as the subordinate;

b) **RCH-on-dialogue**: The RCH-on-dialogue functional unit permits the recovery-context-handles to be varied independently for each dialogue on an association; and

c) **Cancel**: The Cancel functional unit makes use of the CCR functional unit to rapidly pass the rollback semantic through a transaction tree, allowing release of resources from the ready state, without waiting for heuristic or completion messages from lower down the tree. The normal rollback messages are used subsequently.

6.1.3 Dialogue establishment

When a TPSUI attempts to establish a new dialogue, a compatible association must first be assigned for use with this dialogue as described above in 6.1.1.

Should two TPPMs (at different AEIs) attempt to establish a dialogue over the same association (without using the bid mechanism), the TPPM at the contention-winner AEI will succeed, disrupting the attempt from the contention-loser.

Use of the bid mechanism affects dialogue establishment. The contention-loser may request the right to establish a single dialogue without the possibility of a conflict by issuing a bid request. The contention-winner may accept or reject the bid request. If accepted, the contention-loser issues the dialogue establishment request. If rejected, the contention-loser may not issue a dialogue establishment request until after the receipt of a dialogue establishment request from the contention-winner.

Bidding by the contention-loser before attempting to establish a dialogue is mandatory in either of the following cases:

a) if bidding is mandatory for the association;

b) if the condition exists where an unexpected C-BEGIN indication may appear (see 10.3.1).

A dialogue establishment request may be rejected for the following reasons:

a) The TPPM is unable to select or establish an association which meets the requirements of the dialogue.

b) The TPPM is a contention-loser on the association and its dialogue establishment or bid request collides with a dialogue establishment request from the contention-winner. It is worthwhile to distinguish two types of collisions:

   1) The contention-winner is still within a dialogue.

   2) The contention-winner is not within a dialogue. This can happen if the contention-winner begins a dialogue and ends it with no response required from the partner.

c) The partner TPPM or TPSUI rejects the dialogue. This can happen for a variety of reasons: TPSU title not found, insufficient resources, etc. The reason for the rejection is carried on a TP APDU.

NOTE – Although this Recommendation specifies that the dialogue establishment is rejected, this does not preclude an implementation from attempting to retry the dialogue establishment.

Because of the use of unconfirmed dialogue or channel termination, it may happen that "stray APDUs" arrive from the partner after a dialogue establishment request. To detect and discard these stray APDUs, a correlator value is sent on the dialogue establishment request, which is returned by the partner at dialogue establishment confirmation time. As dialogue establishment confirmation always precedes any other request issued by the partner, APDUs received before that confirmation are discarded. (The same mechanism is used during channel establishment.)

The mechanism for detecting the dialogue reject situation described in b) 2) above is the use of a "Last Partner Identifier" (LPI). When the contention-loser issues a bid request or a dialogue establishment request without a bid request, the request carries the correlator of the previous dialogue establishment indication received from the contention-winner. If the contention-winner receives an LPI with a value different from that of the correlator on the previously issued dialogue establishment request, the bid (or dialogue establishment) request is rejected. The LPI is not provided if there was no previous dialogue establishment indication from the contention-winner.

6.1.4 Soliciting dialogue establishment

If the solicit-dialogue functional unit is selected on an association, the contention-loser can solicit the establishment of a dialogue on that association, asking that the dialogue be initiated from the peer. The dialogue, if established, will be between a superior TPSUI at the peer system (the solicited system), and a subordinate TPSUI at the system which solicited the dialogue (the soliciting system). The dialogue solicitation identifies the acceptable TPSU-titles (or set of TPSU-titles) at each side.

If the solicited system is prepared to establish an appropriate dialogue, a TPPM at the peer will receive a dialogue request from an appropriate TPSUI. This will invoke the procedures to assign a compatible association which will assign the association on which the solicitation was received. All subsequent procedures then operate as specified for an unsolicited dialogue, including either the creation of a new TPSUI at the soliciting system or the rejection of the dialogue.

If the solicited system is not prepared to establish a solicited dialogue, it replies rejecting the solicitation.

The solicitation of the dialogue can collide with the establishment of a dialogue or channel on the same association. If it collides with a dialogue request that meets the requirements of the solicitation, the collision is ignored; the solicitation will be considered successful from the viewpoint of the soliciting system and the solicitation message is ignored at the solicited system, without affecting the dialogue. If the collision is with a dialogue request that does not meet the requirements of the solicitation or a channel establishment request, the solicitation is considered to be rejected from the viewpoint of the soliciting system, although no additional service primitive is issued. The solicitation message is ignored at the solicited system, without affecting the dialogue or channel.

A dialogue solicitation cannot be made on an association on which a bid has been accepted.

6.1.5 Channel management

After a node crash or a communications failure, a TPPM may (depending on the transaction state as recorded in a log record found for that transaction) be responsible for recovery. In order to meet this requirement, the TPPM needs a channel for the purpose of recovery. The establishment of channels is in most respects similar to the establishment of dialogues; there is, however, one essential difference: unlike dialogues, channels are not established by the TPPMs themselves, but rather they are established and managed by a Channel Protocol Machine (CPM). There is only a single CPM per AEI and this CPM deals with the channels requested and used by all the TPPMs residing at this AEI.

The interactions between a TPPM and the CPM are modelled by the CAF-service (CAF for Channel Auxiliary Facility). A TPPM uses a CAF service request to request the CPM to establish a channel to a specific partner TPPM. Upon receipt of this request, the CPM either selects an existing channel or establishes a new channel using a procedure similar to that of dialogue establishment (see 6.1.3).

When recovery has been performed, i.e. when a TPPM either has issued a C-RECOVER response or received a C-RECOVER confirm, the TPPM uses another CAF service request in order to inform the CPM that it has no further use for the channel. The CPM then may either terminate the channel or keep it for subsequent use.

Besides satisfying the requests for channels that are issued by TPPMs residing at its AEI, the CPM is also responsible for responding to all channel establishment indications addressed to its AEI and issued by other CPMs. Moreover, the CPM receives all recovery initiating indications on a channel and directs them to the TPPMs to which they are addressed; whenever no such TPPM may be located, the CPM must respond to the recovery initiating indication.

6.1.6 Channel utilization

A channel is established as either a one-way-recovery channel or a two-way-recovery channel. With either type of channel, due to restrictions in ITU-T Rec. X.851 | ISO/IEC 9804, only a single C-RECOVER request is allowed to be outstanding on a channel until it has been responded to; moreover, the issuer of a C-RECOVER request must own the token unless the C-RECOVER request is issued in response to a C-RECOVER indication [C-RECOVER (commit) request in response to a C-RECOVER (ready) indication] or in some circumstances on a two-way-recovery channel.

On a one-way-recovery channel, only the initiator of the channel has the right to initiate recovery. The token, once owned by the initiator, is never transferred to the partner.

On a two-way-recovery channel, either side of the channel may initiate recovery, provided that it owns the token. The token is transferred to the partner after each C-RECOVER request or AF-RECOVER request, unless the C-RECOVER request is issued in response to a C-RECOVER indication or an AF-RECOVER indication; this allows interleaving of recovery exchanges over the channel. If the side that does not have control of the channel wants to initiate recovery on a channel, it may issue an AF-TOKEN-PLEASE request.

6.1.7 Token control

NOTE 1 – See Annex B for U-ASE use of tokens.

CCR requires the Session Layer synchronize-minor token (hereinafter called the token – see 7.3) to be owned when beginning a transaction, committing a transaction, or initiating recovery. The TPPM guarantees that the token will be available at the appropriate times in the absence of the movement of the token by the TPSUI or U-ASE. The TPPM uses the following rules for moving the token:

a) The token is owned by the contention-winner when an association is established.

b) The token is returned to the contention-winner at the termination of the dialogue.

c) If the token is received by a contention-loser while the association is not assigned to a dialogue, it is returned to the contention-winner. This rule does not hold if the contention-loser is attempting to establish a dialogue and has received a confirmation that a bid request was accepted.

   NOTE 2 – This happens when a U-ASE request to move the token collides with an unconfirmed dialogue termination request.

d) The token is moved to the contention-loser upon acceptance of a bid request carrying a parameter requesting the token.

e) The token is moved to the contention-loser upon receipt of a dialogue establishment indication (without prior bidding) selecting the Commit functional unit if the contention-winner owns the token and has not reserved the association for other use. If the token is not owned by the contention-winner, it will eventually arrive and then be sent back to the contention-loser.

   NOTE 3 – This mechanism ensures that the token is always at the dialogue superior when the Commit functional unit is selected to enable the dialogue superior to begin a transaction.

f) If the token is owned when a commitment offer is about to be made, the token is sent when sending the commitment offer.

   NOTE 4 – When, with the dynamic commitment functional unit, there is a collision of commitment offers, the TPPM receiving the token after sending the commitment offer does not transfer it again. When it (almost immediately) receives the commitment offer, since it now has the token, it will make the decision to commit and order commitment on this dialogue.

g) On a two-way-recovery channel, the token is sent to the partner after initiating each recover request.

There are some cases where the token may not be immediately available when required for beginning a transaction or initiating recovery due to the movement of tokens by previous dialogues or other factors. In these cases, except for the case when the U-ASE moves the token within the same dialogue and the token is needed for beginning a transaction, the rules guarantee that the token will eventually arrive, so the TPPM simply waits until it does. A U-ASE that moved the token prior to beginning a transaction is responsible for getting it if it does not have it.

6.1.8 Collisions of ready signals

The resolution of collisions of ready signals is used to continue in a consistent manner at both nodes without exchanging any further messages. This mechanism is primarily based on the observation that a commit decision is reachable without taking both signals into account, i.e. one of these signals can be ignored.

A tie-break mechanism is used for the resolution of the collision of ready signals on a dialogue. The consistent continuation of processing at a single node depends on the ownership of the token:

a) the node which owns the token acts as if no ready signal was sent; the node becomes the commitment coordinator after writing the log-commit record;

b) the node which does not own the token acts as if no ready signal was received; the node is a commit slave and stays in the ready-state until the decision is received.

Following this rule, exactly one of the ready signals will be ignored by both nodes.

The following collisions are not resolved by the tie-break:

a) The collision of ready signals on different channels for a specific transaction branch always leads to rollback.

b) In case of the collision of a ready signal with a one-phase signal or a read-only signal, the ready signal is ignored by both nodes.

c) The collision of a one-phase signal with a one-phase signal or with a read-only signal leads always to commit (though no log-record is written). This collision is not detected by the TPPM but within the CCR-PM and with this the commit decision is propagated.

6.1.9 Concatenation/separation

Concatenation is an optional feature which allows multiple APDUs generated by the TPPM (this includes TP, CCR, ACSE, and U-ASE APDUs) to be mapped onto the User data parameter of a Presentation service primitive, reducing the number of Presentation primitives and optimizing performance. Concatenation is performed by the concatenator part of the SACF and is not included in action sequences (the remainder of the SACF is included in action sequences – see 7.1.3).

When concatenation is not used, the state of the supporting layers is always synchronized with the state of the TPPM because of the one-to-one correspondence between Presentation primitives (which affect the state of the supporting layers) and APDUs (which affect the state of the TPPM). When concatenation is used, it is important to maintain this synchronization between the TPPM and the supporting layers. Therefore, the concatenation rules are constructed such that the APDU directly related to the Presentation primitives causing a state change in the supporting layers is always delivered to the TPPM before any other APDUs in the Presentation primitive (which might not be related to the state change in the supporting layers).

NOTE – An example of this is the C-ROLLBACK-RI CCR APDU. If another APDU was concatenated before this APDU, say, a TP-GRANT-CONTROL-RI APDU, then the following scenario would result. The incoming RS PSDU containing, in the Presentation User data, both APDUs would be processed by Session, resulting in a Session state where an S-RESYNCHRONIZE indication was given and thus a P-RESYNCHRONIZE. The TPPM would, however, process the TP-GRANT-CONTROL-RI APDU first, without seeing the C-ROLLBACK indication. The TPPM could then allow the TPSUI to issue a TP-ROLLBACK request, resulting in an outgoing C-ROLLBACK-RI CCR APDU which is mapped to a P-RESYNCHRONIZE request and thus to an S-RESYNCHRONIZE. This S-RESYNCHRONIZE request is received after the S-RESYNCHRONIZE indication was given previously, resulting in a violation of the Session service-user rules (in some cases). The incoming C-ROLLBACK indication still has not been processed by the TPPM. If nothing could be concatenated before the C-ROLLBACK-RI CCR APDU, then the TPPM would have changed its state to reflect the state change in Session (i.e. process the rollback) before it would accept any other input events from the TPSUI.

Separation involves accepting an incoming Presentation primitive and generating a separate event corresponding to each APDU contained in the User data parameter of the Presentation primitive. The entire Presentation primitive is processed before another Presentation primitive is accepted. While concatenation is an optional feature, separation is mandatory; all concatenation sequences that conform to the rules specified in 10.7.2 shall be supported in incoming Presentation primitives.

6.1.10 Embedding

Where semantics of a combination of APDUs require that they be interpreted as a single unit in order to determine the required action, these APDUs are embedded one within the other. This might arise, for example, when an APDU of the TP-ASE defines added value to a CCR APDU.

NOTE – An example of the use of this principle includes the embedding of TP-PREPARE-RI APDU in C-PREPARE request.

The SACF contains a "router" component which handles TP APDUs embedded in indications and confirms. The router causes a CCR or ACSE indication or confirm not carrying a TP APDU to be directly passed to the MACF (through the applicable SACF procedures). If the CCR or ACSE indication or confirm contains an embedded TP APDU, the router will cause the indication or confirm to be passed to the TP-ASE. The TP-ASE then decodes the TP APDU and passes an AF indication or confirm to the MACF (through the applicable SACF procedures) which expresses the combined semantics of the CCR or ACSE service and TP APDU.

6.2 OSI TP Protocol structure

6.2.1 Components of the PM

The protocol specified in this Recommendation provides the services defined by the OSI TP Service, ITU-T Rec. X.861 | ISO/IEC 10026-2 .

The protocol specification for TP is presented as a TP Protocol Machine (TPPM), supplemented by a Channel Protocol Machine (CPM). Within an AEI, there is one CPM with which TPPMs of the same AEI interact for the purpose of recovery.

Interactions between a TPPM and the CPM are represented by a service called the Channel Auxiliary Facility (CAF) service. The CAF service, provided by the CPM to TPPMs, models the ability for channels to be dynamically attached to and detached from a particular TPPM.

The TPPM and the CPM comprise a collection of SAOs controlled by a MACF.

The structure of the OSI TP Protocol is shown in Figures 1 and 2.

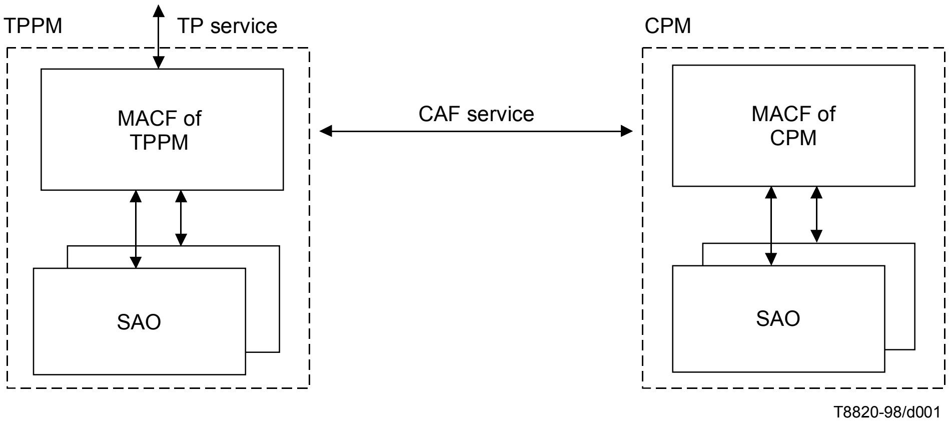
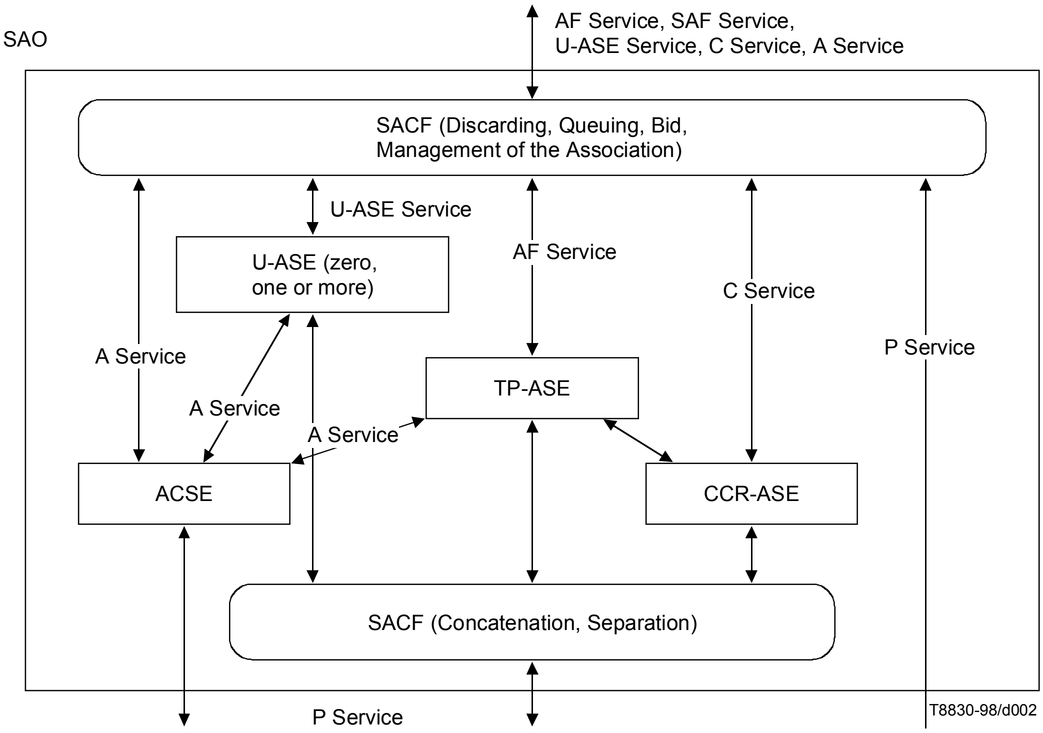


Figure 1/X.862 – Relationship between the TPPM and the CPM


SAO
AF Service, SAF Service, U-ASE Service, C Service, A Service
SACF (Discarding, Queuing, Bid, Management of the Association)
U-ASE Service
U-ASE (zero, one or more)
AF Service
C Service
P Service
A Service
TP-ASE
A Service
A Service
ACSE
CCR-ASE
SACF (Concatenation, Separation)
P Service
T8830-98/d002

NOTE – Only SAOs included in the CPM do not include U-ASE.

Figure 2/X.862 – Structure of the SAO

6.2.1.1 The TPPM

The MACF of the TPPM provides both the TP services over multiple associations and the associated temporal ordering rules. The MACF of the TPPM maps TP services onto the functional capabilities of SAOs included in the TPPM, and onto the CAF service, when appropriate.

Each SAO of a TPPM is composed of:

a) ACSE, to establish and terminate associations – The ACSE services are not invoked directly from the TP Service, but are invoked by the MACF of the TPPM (see 6.1.1 and 8.2), or, in some cases, by some source other than the TPPM MACF procedures [see 11.3.21 (11.3.22 in the case of a CPM) and 10.5.68].

b) TP-ASE, to provide for TP APDU generation and reception – Interactions between the TP-ASE and the MACF (through the applicable SACF procedures) of the TPPM are represented by a service called the Auxiliary Facility (AF) Service.

   The AF service, provided by the TP-ASE, models the ability for TP APDUs to be exchanged and mapped onto appropriate underlying services.

c) CCR, to provide support for commitment, rollback, and recovery functions, when required. CCR services are used by the MACF of the TPPM either:

   1) directly, when no TP APDUs are conveyed by CCR; or

   2) indirectly through the TP-ASE, when TP APDUs are conveyed by CCR (see 8.3).

d) One or more U-ASEs, to provide for application specific protocol – U-ASE services are represented by the TP-DATA service to model TP sequencing constraints on application specific protocols. U-ASE APDUs are mapped onto the Presentation Service, directly or indirectly, as specified by the U-ASE standards.

e) SACF, to coordinate the ASEs in the SAO, in order to maintain a consistent behaviour on the association. The SACF consists of the following components:

   1) Procedures to examine the services passing between the MACF and SAO and handle actions related to dialogue establishment and managing an association while not in use by a dialogue. These actions can be to allow the service primitive to pass through, discard it, generate additional service primitives, or queue the service primitive. In addition, the procedures provide the SACF Auxiliary Facility (SAF) Service, which represents interactions between the MACF and the SACF.

   2) A router which ensures that indications or confirms carrying TP APDUs are directed to the appropriate component. See 6.1.7 on embedding.

   3) A concatenator which optionally constructs a single PSDU based on multiple APDUs for outgoing APDUs, and separates each APDU contained in a PSDU for incoming PSDUs.

Both the AF and the SAF services are purely internal to the TPPM: they are not visible outside of the TPPM, and thus are not available for use by anything other than the MACF of the TPPM. The MACF of the TPPM is the only "user" of both the AF and the SAF services. There shall be no conformance requirements to either the AF service or the SAF service.

An application context for an association to be used by TPPM to assign dialogues shall include:

a) ACSE;

b) TP-ASE;

c) CCR, when the Commit functional unit is selected;

d) one or more U-ASEs, specifically identified. When multiple U-ASEs are included, the relationships between them shall be specified as part of their specification, or as part of the application context specification; and

   NOTE – Annex B contains guidelines for writing U-ASEs eligible for inclusion in a TP application context.

e) SACF and MACF procedures.

6.2.1.2 The CPM

The MACF of the CPM includes the necessary provisions to establish and terminate channels, as appropriate. The MACF of the CPM provides the CAF service. The MACF of the TPPM is the only "user" of the CAF services. There shall be no conformance requirements to the CAF service.

SAOs included in the CPM are the same as those of a TPPM except that no U-ASE is required. The AF and SAF services used by the MACF of the CPM are identical in their definitions to those of a TPPM. They are subject to the same restrictions.

An application context for an association to be used by a CPM to assign channels shall include:

a) ACSE;

b) TP-ASE;

c) CCR; and

d) SACF and MACF procedures.

7 Execution rules

7.1 Operation of the PM

This clause describes the interactions between the components of the PM.

7.1.1 Relationship of SAO(s) to MACF (s)

ITU-T Rec. X.207 | ISO/IEC 9545 defines a one-to-one correspondence between an association and an SAO. While a dialogue or channel is in use on the association, this SAO is associated with an MACF (of a PM). This Recommendation provides for the detaching of the SAO from the MACF at the completion of the dialogue or channel such that the association can subsequently be used by an MACF in another PM. In general, when no dialogue or channel exists on an association, the SAO is detached from the MACF. A detached SAO is not considered to be part of any PM. When an incoming request to establish a dialogue or channel is made, the SAO is then attached to an MACF so that the MACF can process the request. When an MACF initiates the establishment of a dialogue or channel, an SAO that is not attached to any other MACF may be attached to the MACF establishing the dialogue or channel. Alternatively, attempts may be made to create a new SAO to attach to the MACF.

All channels that are not in use for recovery are attached to the CPM. When a request is made of the CPM by a TPPM to perform recovery, the CPM satisfies this request by transferring an unused channel (which is associated with the correct AEI) to the TPPM. When an incoming C-RECOVER indication appears on an unused channel, the CPM finds the TPPM corresponding to the transaction being recovered and transfers the channel to that TPPM. As soon as the TPPM completes recovery, the channel is transferred back to the CPM.

7.1.2 Input events to the PM

PM input events occur either as a result of:

– request and response primitives issued by the TPSUI or indication primitives issued by the CPM in the case of the TPPM; a request issued by a TPPM in the case of the CPM;

– events triggered by local conditions (internal events); or

– events caused by APDUs and Presentation indications and confirmations output from the concatenator part of the SACF or directly from Presentation if concatenation is not used.

7.1.3 Action sequences

An action sequence is a contiguous sequence of execution of procedures in clauses 9, 10, and 11 as a result of a single input event to either the TPPM or the CPM. The CPM and each TPPM have separate action sequences, which proceed independently. An action sequence includes all processing of input events by the PM, except that of the concatenator part of the SACF (see 6.1.9).

An SAO that is not attached to any MACF is not part of any PM action sequences. It may therefore act in parallel with the actions of the PM.

NOTE – This allows association management to be performed independently of normal PM processing.

An action sequence executes completely (i.e. has finished all procedures) before the PM becomes available for handling any subsequent input events.

An action sequence is a serial and atomic execution of the procedures (except in the event of a node crash), with a single exception where parallelism is allowed: when a request or response is issued to the SAO using the (S)AF, ACSE, CCR, or Presentation services, the SAO may continue this action sequence in parallel with the action sequence of the MACF. The SAO ensures that requests or responses issued by the MACF are processed in the order issued. The SAO completes its part of the action sequence when its procedures are finished.

Within an SAO, an A-ABORT request or an A-RELEASE (Result = affirmative) response may be issued. At most one such an event may occur in an action sequence. Such an event is processed by the MACF in order to handle the loss of the association. The event is processed by the MACF in the action sequence in which it was issued once the MACF and all SAOs have completed execution of their procedures (that is, just before the end of the action sequence).

7.1.4 SACF queuing

Certain TP services sometimes cannot be completed in a single action sequence (for example TP-BEGIN-DIALOGUE request). In order to have the TPPM available to accept input events at the TPSUI boundary at all times, it may be necessary to queue requests within the TPPM. Thus, a request that cannot immediately be handled may trigger queuing in the SACF, but is always accepted from the TPSUI. In order to support this, the TPPM describes explicitly queuing operations when they apply.

An example of the need for queuing is that, during dialogue establishment for the contention-loser, a bid may be required before the begin dialogue TP APDU is sent. The begin dialogue AF service is therefore queued by the SACF until the bid is complete. While this queuing is taking place, the TPSUI may make further TP service invocations on this dialogue, which will also be queued.

When it becomes necessary to queue, the queuing takes place within the SACF. Certain AF and CCR services are subject to queuing, in that these services may in certain circumstances be queued for a period of time by the SACF before being issued to the TP or CCR ASE. A queue is established for a single dialogue. This queue may subsequently be fully or partially flushed, which causes all or some of the pending service primitives to be processed in a single action sequence, or it may be discarded, which causes the pending service primitives to be discarded. When a queue is fully flushed, any subsequent service primitives are no longer made pending.

7.1.5 Input event blocking at the PSAP

When completing processing for a transaction, APDUs related to a subsequent transaction may be received on an association. The corresponding service primitives cannot be issued to the TPSUI until the transaction is complete.

When service primitives from an association cannot be processed, the TPPM stops accepting service primitives at the PSAP. When processing can resume, the TPPM resumes accepting service primitives at the PSAP including those previously blocked.

NOTE – The concatenator/separator part of the SACF is not prevented from delivering service primitives to the rest of the SAO while service primitives are blocked at the PSAP. These service primitives would be the result of a PSDU already received by the SACF prior to blocking input events at the PSAP. The concatenation rules are such that these incoming events may be processed correctly in that case.

The TPPM stops accepting input events at the PSAP under the following conditions:

a) when a commit confirm has been received from a subordinate and the dialogue has neither been aborted nor had a deferred end dialogue;

b) when a rollback response is issued to or a rollback confirm is received from a subordinate, the Unchained Transactions functional unit is selected, and the dialogue has not been aborted (by the TPSUI);

c) when a C-BEGIN indication is received in the DECIDED (rollback) state and a TP-DONE request is owed;

d) when a rollback confirm is received from or a rollback response is issued to the superior, the Unchained Transactions functional unit is selected, and a TP-DONE request is owed.

The TPPM resumes accepting input events at the PSAP under the following conditions:

a) when a TP-COMMIT-COMPLETE indication is issued, for each coordinated subordinate dialogue;

b) when a TP-ROLLBACK-COMPLETE indication is issued, for each coordinated subordinate dialogue on which the Unchained Transactions functional unit is selected;

c) when a TP-ROLLBACK-COMPLETE indication is issued, for the superior dialogue;

d) when a TP-U-ABORT request is received on a dialogue on which a rollback response was issued or a rollback confirm was received, and the Unchained Transactions functional unit is selected.

7.1.6 PM error conditions

This Recommendation defines three types of error conditions which may occur during the operation of a PM:

a) *Protocol error* – A condition as described in 7.2 e). The association is aborted when a protocol error occurs. When a protocol error occurs, the relevant Protocol Error procedure (see 10.5.64, 11.3.21, 11.3.22, or 11.3.23) is invoked.

b) *Internal error* – A condition where a local decision is made that the operation of a particular dialogue or channel cannot continue normally. When an internal error on a dialogue is detected, the procedure "Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm on a dialogue" (see 11.3.21) is invoked. When an internal error on a channel owned by the TPPM is detected, the procedure "Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm on a channel" (see 11.3.22) is invoked. When an internal error on a channel owned by the CPM is detected, the procedure "Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm (CPM)" (see 11.3.23) is invoked.

NOTE 1 – Irregularities in the APDU exchanges between U-ASEs are not visible to the TPPM unless they are intentionally signalled to the MACF; in such a case they may be treated as a TP internal error. Although the U-ASE may define such an irregularity as a protocol error, it is not a protocol error in the scope of this Recommendation.

c) *Node crash* – A condition where the operation of the PM cannot continue as determined by local considerations. Upon restart after a node crash, the internal event "Restart after node crash" (see 11.4.3) is triggered.

An internal error or node crash may be triggered by an unexpected event which is received from the TPSUI. Note that rule f) in 7.2 specifies that the procedures assume the operation of the TPSUI is correct according to the service rules, but the state tables in Annex A check for violations of these rules. Should such a violation occur, either an internal error or node crash may be triggered, depending on the scope of the violation as determined locally.

NOTE 2 – An example of a TPSUI triggering a node crash may occur when an error is made in a TP service which affects all dialogues, such as the TP-COMMIT request.

NOTE 3 – A timer expiring is an example of a condition which may trigger an internal error.

7.2 Procedure rules

The following rules govern the execution of the procedures:

a) *Parameter inheritance*

The tables of 9.3 use the notation "(=)" to represent matching of parameters/fields values from the values of parameters/fields of the preceding (causal) primitive/APDU. This value matching is assumed implicitly, so that explicit calling out of the setting of these values is not done in the procedures.

b) *Procedure atomicity*

Each procedure executes atomically, except in the case of a node crash (see 7.1.3).

c) *Inspection of transaction-related state*

The transaction-related state consists of the transaction state, the node type (that is, root, intermediate, or leaf), and whether the node is a superior, subordinate, or does not exist in the transaction tree. Any inspection of the transaction-related state within the procedures refers to the state at the time of the input event which triggered the current action sequence, with one exception: if the MACF is being re-entered in the same action sequence as a result of an A-ABORT request or an A-RELEASE (Result = affirmative) response, inspection of the transaction-related state refers to its value at the time the MACF is re-entered.

NOTE 1 – This handles the situation where the MACF transitioned from the ACTIVE state to the DECIDED (rollback) state, and then the U-ASE issued an A-ABORT request (for example). The state inspection when the MACF is re-entered will not cause a rollback, since the MACF will see the DECIDED (rollback) state and not the ACTIVE state.

d) *Procedure sequence*

The execution of each action specified in the procedure is assumed to be in the sequence specified in the procedure, unless explicitly stated otherwise.

e) *Protocol error detection*

– A protocol error is declared when one of the following conditions occurs:

1) a procedure, other than an internal event procedure (refer to 10.6 and 11.4) is invoked and:

i) no action is taken and no condition is applicable as a result of that procedure invocation; or

NOTE 2 – The action "continue" is used in the procedures as a null action, to avoid incurring errors of this type. Other conditions in the same procedure may be applicable, however, resulting in non-null actions being taken.

ii) the procedure precondition is violated. (Preconditions are presented at the beginning of some procedures in the form of declarative sentences.);

2) a TP indication or confirm is issued that violates the constraints specified in ITU-T Rec. X.861 | ISO/IEC 10026-2;

3) a CCR or ACSE protocol error occurs;

4) a TP APDU is received by the PM which is not encoded as specified in 12.1 "Abstract Syntax of the TP-ASE APDUs", or is not carried on the CCR, ACSE, or Presentation services as specified in 9.5 and 10.7.1;

5) an invalid concatenation sequence is received (see 10.7.2) and is detected as invalid.

NOTE 3 – It is not required that invalid concatenation sequences are detected.

– When a protocol error is declared:

1) If the protocol error occurs in the TP-ASE or SACF with no attached MACF, procedure "Protocol error" (see 10.5.64) is executed.

2) If a TPPM MACF is attached to the association on which the protocol error occurred and a dialogue is active, the procedure "Protocol error or ... on a dialogue" (see 11.3.21) is executed with an indication of a protocol error. As a local decision, this procedure may be invoked on multiple associations, if the protocol error is the result of the attempted issuance of a TP indication or confirm not associated with a particular dialogue.

3) If a TPPM MACF is attached to the association on which the protocol error occurred and a channel is active, the procedure "Protocol error or ... on a channel" (see 11.3.22) is executed with an indication of a protocol error. As a local decision, this procedure may be invoked on multiple associations, if the protocol error is the result of the attempted issuance of a TP indication or confirm not associated with a particular dialogue.

4) If a CPM MACF is attached to the association on which the protocol error occurred, the procedure "Protocol error or ... (CPM)" (see 11.3.23) is executed with an indication of a protocol error.

f) *Service rules assumption*

Each TP service primitive is assumed to have been received in accordance with the constraints and conditions specified in ITU-T Rec. X.861 | ISO/IEC 10026-2 and the requirements in Annex F.

g) *No superior/subordinate assumption*

If an action refers to a superior or subordinate, and there is no superior or subordinate (because the node is a root or leaf, respectively), the action is not taken.

h) *Bounds of the past*

In certain cases, the text refers to a TP or CCR service which has been received. For CCR and TP and related AF services contained in the Commit and Unchained Functional Units, this means that the service primitive was received since the beginning of the current transaction. For all other services, this means that the service primitive was received since the beginning of the current dialogue. In any case, this type of memory of previously received service primitives never survives a node crash.

The terms *ready signal was sent* and *ready signal was received* refer to whether the CCR primitive event (issue of C-READY request, receipt of C-READY indication) occurred since the beginning of the current transaction, unless there has been a node crash. Following a node crash, they refer to whether the event was recorded for the current transaction in the log-record.

i) *Dialogue assumption*

Any dialogue or channel referred to in a procedure is assumed to be the dialogue or channel on which the service primitive was received/issued, unless stated otherwise.

j) *Tree topology terminology*

Whenever the terms superior, subordinate, root node, intermediate node, or leaf node occur without the qualification "dialogue", they are understood to refer to the transaction tree.

7.3 Definitions

When the following definitions or their logical negations are used in the text, they will appear in bold and italics. AF services used in these definitions are defined in 9.2.

7.3.1 ***an AF-TOKEN-PLEASE request is outstanding***: When the CPM has issued an AF-TOKEN-PLEASE request, a subsequent C-RECOVER, AF-RECOVER, or AF-TOKEN-GIVE (two-way-recovery) indication has not been received, and the channel still exists.

7.3.2 ***attach***: The specified SAO and association becomes part of the PM. Any indications or confirms made from that SAO are seen by the PM; the PM may direct requests or responses to the SAO.

7.3.3 ***CAF-PLEASE request is outstanding***: Once issued, a CAF-PLEASE request remains outstanding for a channel until one of the following occurs:

– A CAF-GIVE indication is issued by the CPM for the channel;

– A CAF-FAIL indication is issued by the CPM for the channel;

– A search for the TPPM corresponding to the "Atomic Action Identifier", "Branch Identifier", and "Superior" parameters of the CAF-PLEASE request fails to find the TPPM; or

– A further CAF-PLEASE request is received with the same parameters.

NOTE – A further CAF-PLEASE request with the same parameters can be received following a node crash.

7.3.4 ***basic functional units***: The following functional units are collectively referred to as the *basic functional units*: Dialogue, Shared Control, Polarized Control, Handshake, Commit, Chained Transactions, Unchained Transactions and Recovery. If the Functional-Units-Capability field is not present on the TP-INITIALIZE-RI/-RC exchange, the *basic functional units* are assumed to be available for selection.

NOTE – In practice, they may not be, but the first edition of this Protocol Specification did not provide for identification of functional unit capability at association establishment.

7.3.5 ***carrying the reporting status***: Various auxiliary request and response service primitives are required to be issued *carrying the reporting status.* These primitives always have a Heuristic-Report parameter and, in most cases, Severity, Diagnostic and Completion-data parameters. *Carrying the reporting status* means these parameters shall be set as follows:

a) the Heuristic-Report parameter shall be absent if the Heuristic Containment Required functional unit is selected on the dialogue, or on the dialogue, now detached, that supported the branch; it shall be present otherwise;

b) if present, according to a), the Heuristic-Report parameter shall be set to the current value of the log-damage record, if there is a log-damage record, or set to "none" if there is no log-damage record;

c) the Severity, Diagnostic and Completion-data parameters shall be absent if the Completion Diagnostics functional unit is not selected on the dialogue, or if the primitive is issued on a channel;

d) if the primitive is issued on a dialogue on which the Completion Diagnostics functional unit is selected, and no TP-EARLY-EXIT request was received, the Completion-data parameter on the primitive shall be set to the value of the Completion-data parameter on the most recent TP-DONE request on which the parameter was present (this may be a null value);

e) if the primitive is issued on a dialogue on which the Completion Diagnostics functional unit is selected and the TPPM is in the DECIDED (rollback) state and no TP-EARLY-EXIT request was received, the Severity parameter on the primitive shall be set to the value of the Severity parameter on the most recent TP-DONE request on which the Severity parameter was present;

f) if the primitive is issued on a dialogue on which the Completion Diagnostics functional unit is selected and the TPPM is in the DECIDED (rollback) state, and a *rollback indication* was not received on the dialogue, the Diagnostics parameter of the primitive shall set to the first applicable value of the following:

    1) "user-rollback" if a rollback initiating request was received and no TP-ROLLBACK indication was issued;

    2) "user-data-transaction-completion-collision" if a TP-ROLLBACK indication with that value in the Diagnostic parameter was issued;

    3) "early-exit-transaction-completion-collision" if a TP-ROLLBACK indication with that value in the Diagnostic parameter was issued;

    4) the value from the Diagnostic parameter on any TP-COMPLETION-REPORT indication issued with the Diagnostic parameter; it is a local option which value is used if there is more than one such TP-COMPLETION-REPORT indication;

    5) "other-provider-rollback", if none of the previous conditions apply;

g) if the primitive is issued on a dialogue on which the Completion Diagnostics functional unit is selected, a TP-EARLY-EXIT request was received and a C-ROLLBACK indication was received, the Completion-data parameter on the primitive shall be set to the value of the Completion-data parameter on the TP-EARLY-EXIT request;

h) if the primitive is issued on a dialogue on which the Completion Diagnostics functional unit is selected, a TP-EARLY-EXIT request was received and a C-ROLLBACK indication was received, the Severity parameter on the primitive shall be set to the value of the Severity on the TP-EARLY-EXIT request.

If no value is assigned to a parameter by the above conditions, the parameter shall be absent.

NOTE – The phrase is used only for primitives issued on the superior dialogue, or in the recovery sequence for a branch on the former superior dialogue.

7.3.6 ***close the PSAP***: No further PSDUs are accepted at the PSAP unless and until the PSAP is opened (see 7.3.33, "Open the PSAP").

7.3.7 ***commit confirm***: One of the following:

- C-COMMIT confirm;
- C-RECOVER (done) confirm;
- AF-REPORT (commitRC) indication;
- AF-REPORT (recoverDoneRC) indication;
- AF-ABORT (user, commitRC) indication; or
- AF-ABORT-AND-REPORT (commitRC) indication.

7.3.8 ***commit indication***: One of the following:

- C-COMMIT indication;
- C-COMMIT+C-BEGIN indication;
- AF-ABORT (user, commitRI) indication;
- C-RECOVER (commit) indication; or
- AF-RECOVER (commit) indication.

7.3.9 ***commit response***: One of the following:

- C-COMMIT response;
- AF-REPORT (commitRC) request;
- AF-ABORT (user, commitRC) request; or
- AF-ABORT-AND-REPORT (commitRC) request.
- C-RECOVER (done) response.

7.3.10 ***commit request***: One of the following:

- C-COMMIT request;
- C-COMMIT+C-BEGIN request;
- AF-ABORT (user, commitRI) request;
- C-RECOVER (commit) request; or
- AF-RECOVER (commit) request.

7.3.11 ***DECIDED (commit-one-phase) state***: The node is in the DECIDED (commit) state as specified in ITU-T Rec. X.860 | ISO/IEC 10026-1, but has previously been in the ONE-PHASE state and not in the READY state.

NOTE – This Protocol Specification uses the term "TPPM is in the DECIDED (commit) state" only when the TPPM has previously been in the READY state.

7.3.12 ***detach***: The specified SAO and association ceases being part of the PM. The SAO is no longer able to make indications or confirms to any PM; no PM may direct requests or responses to the SAO.

7.3.13 ***dialogue has been detached***: One of the following service primitives has been issued/received by the PM for the dialogue:

- SAF-DETACH-ASSOCIATION request;
- SAF-ASSOCIATION-LOST indication;
- A-ABORT request;
- A-[P-]ABORT indication;
- A-RELEASE (Result = affirmative) response or confirm.

7.3.14 ***dialogue is chaining transaction branches; chaining dialogue***: When all of the following conditions are met:

- the dialogue exists;
- the Chained Transactions functional unit was selected;
- no TP-U-ABORT request was received for the dialogue;
- no AF-ABORT, AF-ABORT-AND-REPORT, or A[-P]-ABORT indication was received for the dialogue; and
- if the outcome of the transaction is either commitment or unknown, no TP-DEFERRED-END-DIALOGUE request and no AF-DEFER (end-dialogue) indication has been received for the dialogue, unless an AF-EARLY-EXIT request has been issued or AF-EARLY-EXIT indication received.

7.3.15 ***dialogue supports a continuing branch***: The dialogue has a coordination level of "commitment" and no C-NOCHANGE indication or AF-NOCHANGE indication or AF-EARLY-EXIT indication has been received.

7.3.16 ***dialogue will be coordinated***: A dialogue established with the following:

- the Chained Transactions functional unit; or
- the begin-transaction parameter is set to "true" on the TP-BEGIN-DIALOGUE request or indication.

NOTE – This definition is necessary because the coordination level does not become defined until the TP-BEGIN-DIALOGUE indication is made. This definition is intended only for use during dialogue establishment.

7.3.17 ***discard a queue***: The SACF discards all queued service primitives.

NOTE – This definition is used only when a dialogue is terminated.

7.3.18 ***discard any PDUs in the separator***: The SACF discards all queued PDUs in the separator.

7.3.19 ***establish a queue***: The SACF ensures that the AF, CCR, and Presentation service primitives invoked subsequent to the establishment of the queue are queued.

7.3.20 ***flush a queue***: The SAO processes some or all (depending on the context in which the definition is used) of the queued service primitives in the order in which they were queued. If all requests and responses were flushed, the SACF then ensures that any subsequently invoked AF, CCR, ACSE, and Presentation requests and responses are not queued.

NOTE – The processing of all requests and responses on the queue is done by the SAO atomically as part of the action sequence in which the queue is flushed.

7.3.21 ***forget a transaction***: Remove from secure storage the log-ready or log-commit record concerning this transaction, retaining, if any, the log-heuristic and/or the log-damage records.

7.3.22 ***heuristic reporting applies on a dialogue/on a branch***: For a dialogue, the Heuristic Containment Required functional unit was not selected when the dialogue was established; for a branch, the Heuristic Containment Required functional unit was not selected on the dialogue (possibly now *detached)* that originally supported the branch.

7.3.23 ***identifies a branch***: A CCR or AF service primitive that includes parameters Atomic Action Identifier and Branch Identifier *identifies* the corresponding branch.

7.3.24 ***intermediate log-record has been rewritten***: A *commit indication* was received and a log-commit record is in secure storage [as a result of invoking internal event "Rewriting intermediate record" (see 11.4.9)].

7.3.25 ***last commit confirm was received***: When all of the following conditions are met:

- the TPSUI does not *owe a TP-DONE request*;
- a *commit confirm* has been received from each neighbour to which a *commit request* was issued.

7.3.26 ***last partner identifier is valid***: The last-partner-identifier parameter of the most recently received AF-BEGIN-DIALOGUE indication or AF-BID indication is either:

- absent or may carry any value, if no AF-BEGIN-DIALOGUE request was issued on this association; or
- has a value equal to the Correlator of the most recently issued AF-BEGIN-DIALOGUE request on this association.

7.3.27 ***last ready was received***: When all of the following conditions are met:

- at least one of the following is true:
  i) TP-COMMIT request has been received; or
  ii) TP-READ-ONLY request or TP-ONE-PHASE request has been received and there is *TPPM bound data*;
  iii) TP-READ-ONLY request or TP-ONE-PHASE request has been received and more than one C-READY indication has been received;

and

- if the TPPM previously started to set the *TPPM bound data* into the ready-to-commit state, the *TPPM bound data* are in the ready-to-commit state; and

  NOTE – The TPPM may either atomically set the bound data to the ready-to-commit state when writing the log-ready record, or may start to set the bound data to the ready-to-commit state before writing the log-ready record. This condition applies only to the case where the TPPM previously started to set the bound data to the ready-to-commit state.

- a C-READY indication or *ready-substitute indication* has been received on all branches of the transaction.

7.3.28 ***last ready-substitute has been received***: When all of the following conditions are met:

- a TP-ONE-PHASE request or TP-READ-ONLY request has been received;
- *there is no bound data at the TPPM*;
- a *ready-substitute indication* has been received on all branches of the transaction.

7.3.29 ***last rollback confirm was received***: When all of the following conditions are met:

– the TPSUI does not *owe a TP-DONE request*;

– a *rollback indication* or *rollback confirm* has been received from each of the subordinates whose *dialogue has not been detached*.

7.3.30 ***lazy-log-forget is applicable***: The "lazy log forgets tolerated" functional unit was available on the association that supported the dialogue that supported the branch on which a *commit request* was issued and there is no log-commit record and bound data will not be released in the initial state even if there were a node crash followed by the procedures for restart after node crash.

NOTE – Performing a lazy log forget implies that the TPPM may, following a node crash, issue a C-RECOVER (Ready) request for a transaction branch after it has issued a *commit-confirm.* In response, the TPPM may receive a C-RECOVER (unknown) confirm or a C-RECOVER (commit) indication, depending on the state of the partner. The C-RECOVER (unknown) confirm implies that the transaction has rolled back. The local decision of performing a lazy log thus requires that the TPPM guarantees that the ACID properties of the transaction will not be violated in this case.

7.3.31 ***non-recovery states may be entered***: When all of the following conditions are met:

– a TP-ONE-PHASE request or TP-READ-ONLY request has been received;

– *there is no bound data at the TPPM*;

– there is precisely one dialogue of the transaction on which no *ready-substitute indication* has been received and on that dialogue:

   i) *ready-is-sendable*; and

   ii) at least one of the following is true:

      a) an AF-PREPARE indication has been received; or

      b) it is a subordinate dialogue and no AF-PREPARE (data-permitted) request has been sent; or

      c) it is a superior dialogue and the Implicit Prepare functional unit is selected and no AF-PREPARE (data-permitted) request has been sent;

– no C-READY indication has been received on any other dialogue.

NOTE – *Non-recovery states may be entered* is true even if a C-READY indication has been received on the one branch of the transaction on which a *ready-substitute indication* has not been received.

7.3.32 ***one-phase indication***: One of the following:

– C-NOCHANGE indication;

– AF-NOCHANGE indication.

7.3.33 ***open the PSAP***: PSDUs are now accepted at the PSAP.

7.3.34 ***ready signal was sent (to a neighbour/on a dialogue)***: A log-ready record was written identifying that a C-READY request will be issued (to the neighbour/on the dialogue) and (in the same action, unless there was a failure) a C-READY request was so issued.

7.3.35 ***ready signal was received (from a neighbour/on a dialogue)***: A C-READY indication was received (from the neighbour/on the dialogue) since the beginning of the current transaction; or, if there has been a node failure since the beginning of the current transaction, a log-record was written that stated that a C-READY indication was received from the neighbour. The condition remains true till the end of the transaction even if the log-record has since been removed.

7.3.36 ***ready-is-receivable (on a dialogue)***: The dialogue is included in the current transaction and one of the following sets of conditions is true:

i) the dialogue is to a subordinate, the Commit functional unit is selected and the Dynamic Commit functional unit is not selected on the dialogue; or

ii) the dialogue is to a superior and the One-phase commit functional unit is selected and the Commit functional unit is not selected;

   NOTE – In case ii), the dialogue is using static one-phase commit and a ready signal will not be received.

iii) the dialogue is to a subordinate and the Subordinate-may-send-ready parameter on the TP-BEGIN-DIALOGUE request had the value "true";

iv) the dialogue is to the superior and the Superior-may-send-ready parameter on the AF-BEGIN-DIALOGUE indication had the value "true".

7.3.37 ***ready-is-sendable (on a dialogue)***: The dialogue is included in the current transaction and one of the following sets of conditions is true:

i) the dialogue is to the superior, the Commit functional unit is selected and the Dynamic Commit functional unit is not selected on the dialogue; or

ii) the dialogue is to a subordinate and the One-phase commit functional unit is selected and the Commit functional unit is not selected;

NOTE – In case ii), the dialogue is using static one-phase commit and a ready signal will not be sent.

iii) the dialogue is to the superior and the Subordinate-may-send-ready parameter on the AF-BEGIN-DIALOGUE indication had the value "true";

iv) the dialogue is to a subordinate and the Superior-may-send-ready parameter on the TP-BEGIN-DIALOGUE request had the value "true".

7.3.38 ***ready state may be entered***: When all of the following conditions are met:

– a local decision determines that (if the other conditions are met) the READY state should be entered if possible; and

NOTE 1 – It is expected that this decision will usually be to attempt to enter the ready state. The decision not to attempt to enter the READY state is equivalent to the existence of an additional subordinate (or local resource) which is not yet ready.

NOTE 2 – The local decision can be different when "ready state may be entered" is evaluated more than once.

– at least one of the following is true:

  i) TP-COMMIT request has been received; or

  ii) TP-READ-ONLY request or TP-ONE-PHASE request has been received and there is *TPPM bound data*;

  iii) TP-READ-ONLY request or TP-ONE-PHASE request has been received and at least one C-READY indication has been received,

and:

– if the TPPM previously started to set the *TPPM bound data* into the ready-to-commit state, the *TPPM bound data* are in the ready-to-commit state; and

NOTE 3 – The TPPM may either atomically set the bound data to the ready-to-commit state when writing the log-ready record, or may start to set the bound data to the ready-to-commit state before writing the log-ready record. This condition applies only to the case where the TPPM previously started to set the bound data to the ready-to-commit state.

– there is precisely one dialogue in the transaction on which no C-READY indication or *ready-substitute indication* has been received, and, on that dialogue:

  i) none of C-READY request, C-NOCHANGE request and AF-NOCHANGE request have been issued; and

  ii) *ready-is-sendable*; and

  iii) at least one of the following is true:

    a) an AF-PREPARE indication has been received; or

    b) it is a subordinate dialogue and no AF-PREPARE (data-permitted) request has been sent; or

    c) it is a superior dialogue and the Implicit Prepare functional unit is selected and no AF-PREPARE (data-permitted) request has been sent; and

  iv) at least one of the following is true:

    a) it is a superior dialogue; or

    b) an AF-BEGIN-DIALOGUE confirm has been received; or

    c) the RCH-on-dialogue functional unit was not selected on the supporting association; or

    d) when the supporting association was established, on whichever of TP-INITIALIZE-RI or TP-INITIALIZE-RC APDU was received, the Senders-RCH-Varies parameter had the value "false".

NOTE 4 – The last condition, iv), ensures that a log-ready record is not written if it might contain the wrong recovery-context-handle. This will only occur if the superior has begun a dialogue and issued a TP-COMMIT request before any reply has been received from a subordinate, and the subordinate's TPPM is liable to change the recovery-context-handle for each dialogue.

7.3.39 ***ready-substitute indication***: C-NOCHANGE indication or AF-NOCHANGE indication or AF-EARLY-EXIT indication.

NOTE – Although an AF-EARLY-EXIT indication is not equivalent to a 'ready signal without recovery' in the way that C-NOCHANGE and AF-NOCHANGE indication are, it is included in this definition because the effect in the procedures is similar. A dialogue on which AF-EARLY-EXIT indication has been received will only remain in the transaction if the Chained functional unit is selected.

7.3.40 ***ready-substitute request***: C-NOCHANGE request or AF-NOCHANGE request.

7.3.41 ***reporting applies on a dialogue/on a branch***: For a dialogue, the Heuristic Containment Required functional unit is not selected or the Completion Reporting functional unit is selected or both; for a branch whose *dialogue has not been detached*, as for that dialogue; for a branch whose *dialogue has been detached*, only that the Heuristic Containment Required functional unit was selected on the dialogue that originally supported the branch.

A *report is to be sent* when at least one of the following is true; *a report is not to be sent* when all are false:

a) the Heuristic Containment Required functional unit is not selected on the superior dialogue and a log-damage record exists;

b) the Completion diagnostics functional unit is selected on the superior dialogue and, on the most recent TP-DONE request on which the Completion-data parameter was present, that parameter had a non-empty value;

c) the Completion diagnostics functional unit is selected on the superior dialogue and a TP-DONE request has been received on which the Severity parameter was present;

d) the Completion diagnostics functional unit is selected on the superior dialogue and a TP-EARLY-EXIT request was received on which the Completion-data parameter had a non-empty value or the Severity parameter was present and C-ROLLBACK indication was received.

7.3.42 ***reporting status is known***: To a TPPM when:

a) a TP-DONE request is not owed; and

b) for each branch to a subordinate (if there are any) at least one of the following is true:

   i) the Heuristic Containment Required functional unit was selected on the dialogue (possibly now detached) that supported the branch and either the dialogue is detached or the Completion Reporting functional unit is not selected;

   ii) a commit-confirm or rollback-confirm or *ready-substitute indication* has been received;

   iii) an AF-REPORT indication or AF-ABORT-AND-REPORT indication has been received;

   iv) a CAF-RECOVER indication with the Heuristic Report parameter present has been received;

   v) any heuristic condition that may yet be reported will be compensated by the TPPM and either the dialogue is detached or the Completion Reporting functional unit is not selected.

7.3.43 ***rollback confirm***: One of the following:

– C-ROLLBACK confirm;

– AF-EARLY-EXIT confirm;

– AF-REPORT (rollbackRC) indication;

– AF-ABORT (user/provider, rollbackRC) indication;

– AF-ABORT-AND-REPORT (rollbackRC) indication; or

– AF-BEGIN-DIALOGUE (accepted/rejected(user), rollbackRC) confirm.

7.3.44 ***rollback indication***: One of the following:

– C-ROLLBACK indication;

– AF-EARLY-EXIT indication;

– AF-REPORT (rollbackRI) indication;

– AF-ABORT (user/provider, rollbackRI) indication;

– AF-ABORT-AND-REPORT (rollbackRI) indication; or

– AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) confirm.

7.3.45 ***rollback reporting has completed***: A *rollback response* was issued to the superior or a *rollback confirm* was received from the superior, or there is no superior dialogue.

7.3.46 ***rollback request***: One of the following:

- C-ROLLBACK request;
- AF-REPORT (rollbackRI) request;
- AF-ABORT (user/provider, rollbackRI) request; or
- AF-ABORT-AND-REPORT (rollbackRI) request.

7.3.47 ***rollback response***: One of the following:

- C-ROLLBACK response;
- AF-REPORT (rollbackRC) request;
- AF-ABORT (user/provider, rollbackRC) request; or
- AF-ABORT-AND-REPORT (rollbackRC) request.

7.3.48 ***token***: The Session Layer synchronize-minor token which is required by CCR.

7.3.49 ***a TP-DONE request is owed***: when either or both of the following cases holds:

a) the TPPM has received or issued one or more of the following service primitives without having received a subsequent TP-DONE request:

   - a TP-COMMIT indication;
   - a TP-ROLLBACK request/indication;
   - a TP-UNKNOWN indication;
   - a TP-COMPLETION-REPORT indication;

b) the TPPM is in the DECIDED (commit) or DECIDED (rollback) state and has received or issued one or more of the following service primitives on a dialogue with coordination level of "commitment" without having received a subsequent TP-DONE request:

   - a TP-P-ABORT indication;
   - a TP-U-ABORT request/indication; or
   - a TP-BEGIN-DIALOGUE (rejected) confirm.

7.3.50 ***TPPM bound data***: The bound data controlled by the TPPM.

7.3.51 ***transaction initiation purging period***: A state of a TPPM with respect to a given dialogue which is entered upon receipt of a TP-BEGIN-TRANSACTION request if:

a) the Shared Control functional unit is selected; and

b) there is a *user error purging period*.

The *transaction initiation purging period* is terminated as soon as:

a) the number of AF-U-ERROR confirms, AF-END-DIALOGUE indications with the Confirmation parameter set to "true", and AF-HANDSHAKE indications received since the reception of the TP-BEGIN-TRANSACTION request equals the number of TP-U-ERROR requests that were outstanding at that time; or

b) a *rollback confirm* is received.

7.3.52 ***transfer the channel***: The specified SAO is transferred from the PM to another PM. The specified service invocation and any subsequent service invocations made at this SAO are sent/received to/by the PM to which the SAO is transferred.

7.3.53 ***the two-way-recovery token is expected***: When the CPM has received a C-RECOVER or an AF-RECOVER indication, and the subsequent AF-TOKEN-GIVE (two-way-recovery) indication has not been received.

7.3.54 ***user error purging period***: A state of a TPPM with respect to a given dialogue which is entered upon receipt of a TP-U-ERROR request if:

a) the Polarized Control functional unit is selected and the TPPM does not have control and there is no handshake or dialogue termination indication outstanding; or

b) the Shared Control functional unit is selected and there is no handshake or dialogue termination indication outstanding.

The *user error purging period* is terminated as soon as :

a) if the Polarized Control functional unit is selected and a TP-HANDSHAKE indication, a TP-GRANT-CONTROL indication, a TP-HANDSHAKE-AND-GRANT-CONTROL indication, or a TP-END-DIALOGUE indication is issued; or

b) if the Shared Control functional unit is selected and the number of AF-U-ERROR confirms, AF-END-DIALOGUE indications with the confirmation parameter set to "true", and AF-HANDSHAKE indications received since the beginning of the *user error purging period* equals the number of TP-U-ERROR requests received during that period; or

c) a *rollback confirm* is received.

7.3.55 ***write the log-commit record***: The TPPM ensures that the information specified in 7.4.2 will be available even after the occurrence of a node crash. Remove from secure storage the log-ready record concerning this transaction, if one exists.

7.3.56 ***write the log-damage record***: The TPPM ensures that the information specified in 7.4.4 will be available even after the occurrence of a node crash.

7.3.57 ***write the log-heuristic record***: The TPPM ensures that the information specified in 7.4.3 will be available even after the occurrence of a node crash.

7.3.58 ***write the log-ready record***: The TPPM ensures that the information specified in 7.4.1 will be available even after the occurrence of a node crash.

7.4 Log records used by the PM

The transaction identifier defined in ITU-T Rec. X.860 | ISO/IEC 10026-1 is the atomic action identifier as defined in ITU-T Rec. X.851 | ISO/IEC 9804. The transaction branch identifier defined in ITU-T Rec. X.860 | ISO/IEC 10026-1 is the branch identifier as defined in ITU-T Rec. X.851 | ISO/IEC 9804 .

7.4.1 Log-ready record

The log-ready record contains the following information written in secure storage:

a) to identify the transaction:

- atomic action identifier;

b) to identify the branch to which the *ready signal was sent*:

- branch identifier;
- neighbour's AE Title;
- if provided on the TP-INITIALIZE-RI or -RC APDU, or on the TP-BEGIN-DIALOGUE-RI or -RC APDU, the recovery-context-handle received from the superior;

c) for each neighbour from which a C-READY indication was received, if any, to identify the neighbour:

- branch identifier;
- neighbour's AE Title;
- if provided on the TP-INITIALIZE-RI or -RC APDU, or on the TP-BEGIN-DIALOGUE-RI or -RC APDU, the recovery-context-handle received from the subordinate.

7.4.2 Log-commit record

The log-commit record contains the following information written in secure storage:

a) to identify the transaction:

- atomic action identifier;

b) for each neighbour from which a C-READY indication was received, to identify the neighbour:

- branch identifier;
- neighbour's AE Title;
- if provided on the TP-INITIALIZE-RI or -RC APDU, or on the TP-BEGIN-DIALOGUE-RI or -RC APDU the recovery-context-handle received from the subordinate.

7.4.3 Log-heuristic record

The log-heuristic record contains the following information written in secure storage:

a) the atomic action identifier;

b) the state of the bound data;

c) information necessary to execute compensating actions, if required.

NOTE – The log-heuristic record models the need for an open system to retain, beyond node crashes, information about the heuristic decision.

7.4.4 Log-damage record

The log-damage record contains the following information:

a) the atomic action identifier;

b) the current known state of the bound data in the node's subtree. Its value is either "heuristic-hazard" or "heuristic-mix".

The log-damage record is written in secure storage, except where the node is using one-phase commitment. When using one-phase commitment, the log-damage record is not required to survive node crashes, as there is no recovery.

7.5 Recovery-context-handle

A recovery-context-handle is an identification of a grouping of log records, that will be used for all transactions on a particular association or dialogue. The use of this grouping and associated recovery-context-handle is optional on any particular association or dialogue.

NOTE – A possible use of the recovery-context-handle is to allow the partitioning of the set of log records, each part having a different value for the recovery-context-handle.

When the recovery-context-handle is provided by the partner for a given association or dialogue, its value shall be logged for all transaction branches that are initiated by the partner on this association or dialogue, and, if initiation of recovery is subsequently required for any of these transaction branches, the value of the recovery-context-handle shall be conveyed on the consequent recovery primitives. If values for the recovery-context-handle are passed on both association and dialogue establishment, the value from the dialogue establishment is used for transactions begun on that dialogue. If no value is passed on the dialogue establishment, the value from the association establishment is used, if there is one.

The recovery-context-handle shall only be passed on dialogue establishment if the RCH-on-dialogue functional unit was selected on the supporting association, and on whichever of TP-INITIALIZE-RI and TP-INITIALIZE-RC APDU was sent by the TPPM, the Senders-RCH-Varies parameter had the value "true".

8 Use of ACSE, CCR and the Presentation Layer

8.1 Introduction

This clause identifies the use and requirements made of ACSE, CCR, and the Presentation Layer, and the rules for establishing, assigning, and terminating associations.

8.2 Use of ACSE Service primitives

The Association Control Service Element (ACSE) is used as described in ITU-T Rec. X.217 | ISO/IEC 8649 to establish and release associations.

The TPPM uses the following ACSE services:

- A-ASSOCIATE and A-RELEASE to establish and release associations; and
- A-ABORT to abruptly release an association.

The TPPM also must react to the occurrence of the A-P-ABORT indication primitive.

8.2.1 Use of the A-ASSOCIATE parameters

The parameters of the A-ASSOCIATE service are used by the PMs as specified in Table 1.

Table 1/X.862 – Use of A-ASSOCIATE parameters

A-ASSOCIATE parameters	Used by TPPM	
	Application-supported Transactions	**Provider-supported Transactions or TP Channel**
Mode	Yes	Yes
Application Context Name	Yes	Yes
Calling AP Title	Conditional	Yes
Calling AE Qualifier	Conditional	Yes
Calling AP Invocation Identifier	Conditional	Conditional
Calling AE Invocation Identifier	Conditional	Conditional
Called AP Title	Conditional	Conditional
Called AE Qualifier	Conditional	Conditional
Called AP Invocation Identifier	Conditional	Conditional
Called AE Invocation Identifier	Conditional	Conditional
Responding AP Title	Conditional	Yes
Responding AE Qualifier	Conditional	Yes
Responding AP Invocation-Identifier	Conditional	Conditional
Responding AE Invocation-Identifier	Conditional	Conditional
User Information	Yes	Yes
Result	Yes	Yes
Result Source	Yes	Yes
Diagnostic	Conditional	Conditional
Calling Presentation Address	Yes	Yes
Called Presentation Address	Yes	Yes
Responding Presentation Address	Yes	Yes
Presentation Context Definition List	Yes	Yes
Presentation Context Definition Result List	Yes	Yes
Default Presentation Context Name	No	No
Default Presentation Context Result	No	No
Quality of Service	Yes	Yes
Presentation Requirements	Conditional	Conditional
Session Requirements	Yes	Yes[a)]
Initial Synchronization Point Serial Number	Conditional	Yes
Initial Assignment of Tokens	Conditional	Yes[b)]
Session-Connection Identifier	No	No

[a)] See 8.5.2 and 8.5.3 for constraints on this parameter.

[b)] See 8.5.4 for the assignment of the parameter.

8.2.2 Use of the A-RELEASE parameters

The parameters of the A-RELEASE service are used by the PMs as specified in Table 2.

Table 2/X.862 – Use of A-RELEASE parameters

A-RELEASE parameters	Used by TPPM
Reason	No
User Information	No
Result	Yes

8.2.3 Use of the A-ABORT and A-P-ABORT parameters

The parameters of the A-ABORT and A-P-ABORT services are used by the PMs as specified in Tables 3 and 4.

Table 3/X.862 – Use of A-ABORT parameters

A-ABORT parameters	Used by TPPM
Abort Source	No
User Information	Yes

Table 4/X.862 – Use of A-P-ABORT parameters

A-P-ABORT parameters	Used by TPPM
Provider Reason	No

8.3 Use of CCR Service primitives

The Commitment, Concurrency and Recovery Service Element (CCR) is used for provider-supported transactions.

The PMs use the following CCR services:

– C-BEGIN, C-PREPARE, C-READY, C-COMMIT and C-ROLLBACK for TP services supported by commitment-related functional units; and

– C-INITIALIZE to identify the CCR functional units;

– C-CANCEL for the TP Cancel functional unit; and

– C-NOCHANGE for the one-phase commitment and read-only functional units; and

– C-RECOVER for transaction recovery.

The PMs use the User Data parameter of some CCR services to convey certain TP APDUs. These CCR services and TP APDUs are specified in Table 5.

Table 5/X.862 – Use of C-INITIALIZE parameters

C-INITIALIZE parameters	Used by TPPM
CCR Requirements	Yes
Version	Yes
Ready-collision-reservation	Yes
User Data	No

The parameters of the CCR services are used by the PMs as specified in Tables 6 to 13.

Where the TP procedures refer to the "atomic-action-identifier" parameter of C-BEGIN request and indication, this is the combination of the "Atomic Action Identifier – Owner's Name" and the "Atomic Action Identifier – Suffix". Where the TP procedures refer to the "atomic-action-branch-identifier" parameter of C-BEGIN request and indication, this is the combination of the "Branch Identifier – Branch-owner's Name" and the "Branch Identifier – Suffix".

Table 6/X.862 – Use of C-BEGIN parameters

C-BEGIN parameters	Used by TPPM
Atomic Action Identifier – Owner's Name	Yes
Atomic Action Identifier – Suffix	Yes
Branch Identifier – Branch-owner's name	Yes
Branch Identifier – Suffix	Yes
User Data	Conditional

Table 7/X.862 – Use of C-PREPARE parameters

C-PREPARE parameters	Used by TPPM
User Data	Yes

Table 8/X.862 – Use of C-READY parameters

C-READY parameters	Used by TPPM
User Data	Conditional

Table 9/X.862 – Use of C-COMMIT parameters

C-COMMIT parameters	Used by TPPM
User Data	Conditional

Table 10/X.862 – Use of C-ROLLBACK parameters

C-ROLLBACK parameters	Used by TPPM
User Data	Conditional

Table 11/X.862 – Use of C-CANCEL parameters

C-CANCEL parameters	Used by TPPM
User Data	Conditional

Table 12/X.862 – Use of C-RECOVER parameters

C-RECOVER parameters	Used by TPPM
Recovery State	Yes
Atomic Action Identifier	Yes
Branch Identifier	Yes
User Data	Conditional

Table 13/X.862 – Use of C-NOCHANGE parameters

C-NOCHANGE parameters	Used by TPPM
Confirmation	Yes
Outcome	Yes
User Data	Conditional

8.4 Use of the Presentation Layer

8.4.1 Use of Presentation Service primitives

Implementations shall take care that parallelism between the lower layers and the SAO does not result in a violation of the service user rules of those layers.

NOTE – An example of this problem may occur in the case of a rollback. A C-ROLLBACK indication, which is mapped to a P-RESYNCHRONIZE indication, may be processed by the Session layer and not yet processed by the SAO. Before this indication is processed by the SAO, a C-ROLLBACK request, mapped to a P-RESYNCHRONIZE request may be sent to Session, resulting in a violation of the Session Service user rules.

In addition to Presentation Service primitives used by CCR and ACSE Protocol Machines, the TPPM makes use of:

– P-TOKEN-PLEASE and P-TOKEN-GIVE for token management (to position the tokens correctly for CCR);

– P-DATA for all other services.

The U-ASE may make use of the P-TOKEN-GIVE or P-TOKEN-PLEASE service to manage the Session tokens. In general, the use of these services by the U-ASE is manifested in the TPPM as a TP-DATA request or a U-ASE indication. There are some cases, however, where the P-TOKEN-GIVE indication which refers to the synchronize-minor token may be seen by the PM. In this case, the procedures of the PM will refer to a P-TOKEN-GIVE (sync-minor) request or indication.

The parameters of the Presentation services are used by the TPPMs as specified in Tables 14 to 16.

Table 14/X.862 – Use of P-TOKEN-PLEASE parameters

P-TOKEN-PLEASE parameters	Used by TPPM
Tokens (synchronize-minor)	Yes
User Data	Yes

Table 15/X.862 – Use of P-TOKEN-GIVE (sync-minor) parameters

P-TOKEN-GIVE (sync-minor) parameters	Used by TPPM
Tokens (synchronize-minor)	Yes
User Data	Yes

Table 16/X.862 – Use of P-DATA parameters

P-DATA parameters	Used by TPPM
User Data	Yes

8.4.2 Mapping of C-ROLLBACK-RI to Presentation

CCR requires a C-ROLLBACK-RI to be mapped to a P-RESYNCHRONIZE request. When CCR is used with TP, the Tokens parameter of the P-RESYNCHRONIZE request shall be set so that the synchronize-minor token is passed to the superior. TP places no requirements on the setting of the Tokens parameter for the other available tokens.

8.5 Association management

8.5.1 Introduction

This subclause defines the requirements of this Recommendation with respect to the management and use of associations.

8.5.2 Association/dialogue compatibility

An association is said to be compatible with a dialogue if it meets the following conditions:

a) The association shall have been established with an AEI fulfilling the requirements expressed in the Application Context Name, Recipient-AP-Title, and any of the following for which a value was specified: Recipient-API-Identifier, Recipient-AE-Qualifier, and Recipient-AEI-Identifier parameters as specified by the TP-BEGIN-DIALOGUE request.

b) The association shall have selected the kernel and full duplex Session functional units, the kernel Presentation functional unit, and, if the Commit functional unit is selected, the Session functional units as required by CCR.

   NOTE – This includes the Session Data Separation functional unit required by CCR version 2.

c) If the Commit functional unit is selected, CCR version 2's abstract syntax name shall be found in the Presentation Context Definition List and the application context.

d) The association shall have been established with a Quality of Service parameter compatible with the Quality-of-Service parameter specified by the TP-BEGIN-DIALOGUE request, if any.

e) The TP functional unit capabilities selected for the association include all of the TP functional units selected for the dialogue.

f) The CCR functional units selected for the association shall include all those required to support the TP functional units selected for the dialogue, as listed in Table 17.

8.5.3 Association/channel compatibility

An association is said to be compatible with a channel if it meets the following conditions:

a) the association shall have been established with an AEI identified by the AE-title as specified in the log record(s) for the transaction (s) to be recovered, and with an application context appropriate for recovery;

b) the association shall have selected the kernel and full duplex Session functional units, the kernel Presentation functional unit, and the Session functional units as required by CCR for recovery;

c) CCR version 2's abstract syntax name shall be found in the Presentation Context Definition List and the application context.

8.5.4 Initiating an association establishment

When establishing an association, a TP-INITIALIZE-RI APDU shall be constructed and issued as User Information of the A-ASSOCIATE request. Table 18 lists the fields of this TP APDU.

Table 17/X.862 – TP and CCR functional unit compatibility

If this TP functional unit is selected	These CCR functional units[a] are required
Dialogue	
Shared Control	
Polarized Control	
Handshake	
Commit	Static-commitment or Dynamic-commitment
Chained Transactions	
Unchained Transactions	
Dynamic Commit	Dynamic-commit
Unchecked Tree	
Implicit Prepare	
Read-Only	No-change completion
Early-exit	
One-phase Commit	No-change completion
Completion Diagnostics	
Heuristic Containment Required	
RCH-on-dialogue	
Cancel	Cancel

[a] Where no CCR functional unit is listed in the right-hand column, selection of the TP functional unit does not directly require particular CCR functional units to be selected. The requirements on which TP functional units can be combined means that there are indirect requirements on CCR functional units.

Table 18/X.862 – TP-INITIALIZE-RI/RC APDUs' fields

TP APDU	TP-INITIALIZE-	
Field	RI	RC
Protocol-Version	M	M
Contention-Winner-Assignment	M	
Bid-Mandatory	M	
Recovery-Context-Handle	O	O
Diagnostic		O
Functional-Unit-Capabilities	C	C
Senders-RCH-Varies	O	O

The fields of the TP-INITIALIZE-RI APDU are set as follows:

a) The Protocol-Version field indicates the possible versions of the TP Protocol that can be supported. The TPPM may support more than one version of the protocol.

NOTE 1 – A TPPM may also propose different subsets of the versions it supports; which versions to propose is a local matter.

b) The Contention-Winner-Assignment field identifies whether or not the initiating TPPM will be considered as the contention-winner of this association. This field takes one of the following values:

- "true" (the initiator is the contention-winner);
- "false" (the initiator is the contention-loser).

If CCR is included in the application context of the association, then the following applies to the setting of the A-ASSOCIATE request parameter "Initial Assignment of Tokens":

1) if the value of the Contention-Winner-Assignment field is "true", the value of the A-ASSOCIATE request parameter "Initial Assignment of Tokens" (see 8.2.1) is "requestor side";

2) if the value of this field is "false", the value of the A-ASSOCIATE request parameter "Initial Assignment of Tokens" is "acceptor side".

c) The Bid-Mandatory field indicates whether the use of the bid mechanism (by the contention-loser) is mandatory or not. This field takes one of the following values:

   - "true";
   - "false".

d) The Recovery-Context-Handle field is optionally used to provide a value that is to be supplied when recovery is requested by the remote TPPM.

e) The Functional-Unit-Capabilities field indicates which TP functional units the AEI supports for selection on this association. This value of this field is a set of values, each identifying one of the TP functional units. The set of functional units identified by the value of this field is not required to be a valid selection of functional units for the TP-BEGIN-DIALOGUE service as specified in ITU-T Rec. X.861 | ISO/IEC 10026-2. The Functional-Unit-Capabilities field can be absent if the functional units supported include only some or all of the *basic functional units*.

   NOTE 2 – An association, once established, may be required to support dialogues with different requirements, and also TP channels. Therefore, the requirements declared when the association is established must be sufficient for all intended uses of the association. For example, a Recovery-Context-Handle may be specified even though initial use of the association may be for a dialogue without the Commit functional unit or for a TP channel, neither of which have any use for a Recovery-Context-Handle.

f) The Senders-RCH-Varies field indicates whether the Recovery-Context-Handle field will be present in any TP-BEGIN-DIALOGUE-RI and TP-BEGIN-DIALOGUE-RC APDUs issued by the initiating TPPM. It is only present if the RCH-on-dialogue functional unit is selected in the Functional-Unit-Capabilities field. This field takes one of the following values:

   - "true";
   - "false".

If the Functional-Unit-Capabilities field of the TP-INITIALIZE-RI APDU indicates TP functional units that require CCR functional units other than static-commitment, a C-INITIALIZE request shall be issued with the following parameter values:

a) The CCR Requirements parameter shall indicate at least the CCR functional units required, according to Table 17, by the TP functional units indicated in the Functional-Unit-Capabilities field of the TP-INITIALIZE-RI APDU.

b) The Ready-collision-reservation parameter shall have the value "true".

8.5.5 Receiving an association establishment indication

Upon receipt of an A-ASSOCIATE indication, a TP-INITIALIZE-RI APDU, shall be received as User Information of the A-ASSOCIATE indication. Table 18 lists the fields of this TP APDU.

A C-INITIALIZE indication may also have been issued by the CCR ASE.

The fields of the TP-INITIALIZE-RI APDU are used as follows:

a) If the Protocol-Version field contains a version that is supported by the TPPM, the association may be accepted. If not, the association shall be rejected. The TPPM ignores any values that indicate a later version of the protocol that it can support.

b) If the value of the Contention-Winner-Assignment field is acceptable to the TPPM, the association may be accepted. Otherwise, the association shall be rejected.

c) If the value of the Bid-Mandatory field is acceptable to the TPPM, the association may be accepted. Otherwise, the association shall be rejected.

d) If the Recovery-Context-Handle field is present, its value shall be stored in the recovery log (the log-ready record or the log-commit record, whichever applies, see 7.5) for all transactions on the association (except transactions on dialogues where a different Recovery-Context-Handle is received at dialogue establishment).

e) If the Functional-Unit-Capabilities field is present, values that indicate functional units that are not supported by the TPPM, including functional units not known to the TPPM, are ignored.

If CCR will be used on the association (for a coordinated dialogue or a channel), and the A-ASSOCIATE indication either:

i) does not contain an entry for CCR version 2's abstract syntax name in the Presentation Context Definition List parameter; or

ii) the Session functional units required by CCR version 2 are not selected in the Session Requirements parameter,

then the association shall be rejected. Otherwise, the association may be accepted.

If a C-INITIALIZE indication was issued by the CCR ASE, any TP functional units indicated in the Functional-Unit-Capabilities field that require CCR functional units that are not indicated in the CCR Requirements parameter of the C-INITIALIZE indication shall be ignored. If no C-INITIALIZE indication was issued by the CCR ASE, any TP functional units indicated in the Functional-Unit-Capabilities field that require CCR functional units other than static-commitment shall be ignored.

8.5.6 Responding to association establishment

When responding to association establishment, a TP-INITIALIZE-RC APDUshall be constructed and issued as User Information of the A-ASSOCIATE response. Table 18 lists the fields of this TP APDU.

If a C-INITIALIZE indication was issued by the CCR ASE when the A-ASSOCIATE indication was received, and the association is to be accepted, a C-INITIALIZE response shall be issued to the CCR ASE.

The fields of the TP-INITIALIZE-RC APDU (and the Result parameter of the A-ASSOCIATE response and the parameters of the C-INITIALIZE response, if one is issued) are set as follows:

a) If the association is accepted:

   1) The Result parameter of the A-ASSOCIATE response shall be set to "accepted".

   2) The Protocol-Version field of the TP-INITIALIZE-RC APDU shall be set to the version of ITU-T Rec. X.860-series | ISO/IEC 10026 to be used for this association. This version shall be one of the protocol versions proposed on the TP-INITIALIZE-RI APDU.

   3) The Recovery-Context-Handle field of the TP-INITIALIZE-RC APDU may (optionally) be set to a value that shall be used when recovery is requested by the remote TPPM.

   4) The Diagnostic field of the TP-INITIALIZE-RC APDU shall be omitted.

   5) The Functional-Unit-Capabilities field of the TP-INITIALIZE-RC APDU shall be omitted if the field was omitted on the TP-INITIALIZE-RI APDU. It shall contain the identifications of all the functional units that were included in the field on the TP-INITIALIZE-RI APDU, were not ignored and are supported by the TPPM for selection on this association.

   6) If the Functional-Unit-Capabilities field is present in the TP-INITIALIZE-RC APDU, subsequent TP-BEGIN-DIALOGUE-RI APDUs issued on this association shall not select functional units that are not identified in this field.

   7) If the Functional-Unit-Capabilities field is absent from the TP-INITIALIZE-RC APDU, subsequent TP-BEGIN-DIALOGUE-RI APDUs issued on this association shall not select functional units that are not *basic functional units*.

   8) The CCR requirements parameter of the C-INITIALIZE response shall indicate those CCR functional units required, as specified in Table 17, by the TP functional units identified in the Functional-Unit-Capabilities field of the TP-INITIALIZE-RC APDU; if the Functional-Unit-Capabilities field is absent from the TP-INITIALIZE-RC APDU, and a C-INITIALIZE response is issued to the CCR ASE, it shall identify at least static commitment.

   9) The Ready-collision-reservation parameter of the C-INITIALIZE response shall have the value "true".

   10) If the RCH-on-dialogue functional unit is selected in the Functional-Unit-Capabilities field, the Senders-RCH-Varies field indicates whether the Recovery-Context-Handle field will be present in any TP-BEGIN-DIALOGUE-RI and TP-BEGIN-DIALOGUE-RC APDUs issued by the responding TPPM. This field takes one of the following values:

      - "true";

      - "false".

b) If the association is rejected:

   1) The Result parameter of the A-ASSOCIATE response shall be set to:

      i) "rejected(permanent)", if the Diagnostic field of the TP-INITIALIZE-RC APDU contains any of the settings:

         – "tp-protocol-version-incompatibility";

         – "ccr-version-2-not-available";

      ii) "rejected(transient)", otherwise.

   2) The Protocol-Version field of the TP-INITIALIZE-RC APDU shall be set to the version of ITU-T Rec. X.860-series | ISO/IEC 10026 that can be supported.

   3) The Recovery-Context-Handle field of the TP-INITIALIZE-RC APDU shall be omitted.

   4) The Diagnostic field of the TP-INITIALIZE-RC APDU shall be set to all applicable values of the following:

      i) "ccr-version-2-not-available", if CCR is required on the association and either or both the CCR Version 2 is not available or the Session functional units required for CCR Version 2 are not selected in the A-ASSOCIATE indication;

      ii) "tp-protocol-version-incompatibility", if none of the values of the Protocol-Version field offered in the TP-INITIALIZE-RI APDU can be supported for this association;

      iii) "contention-winner-assignment-rejected", if the value of the Contention-Winner-Assignment field of the TP-INITIALIZE-RI APDU is not acceptable for this association;

      iv) "bid-mandatory-value-rejected", if the value of the Bid-Mandatory field of the TP-INITIALIZE-RI APDU is not acceptable for this association;

      v) "no-reason-given", if none of the above Diagnostic values applies.

   5) The Functional-Unit-Capabilities field shall either be omitted, or shall contain the identifications of the functional units that are supported by the TPPM. It shall be omitted if the field was not present on the TP-INITIALIZE-RI APDU.

8.5.7 Receiving confirmation of association establishment

Upon receipt of an A-ASSOCIATE confirm, a TP-INITIALIZE-RC APDU shall be received as User Information of the A-ASSOCIATE confirm. Table 18 lists the fields of this TP APDU.

The fields of the TP-INITIALIZE-RC APDU are used as follows:

a) If the association is accepted, Protocol-Version defines the protocol version of ITU-T Rec. X.860-series | ISO/IEC 10026 to be used for this association. If the value of the Protocol-Version field is not one of the versions that was proposed on the TP-INITIALIZE-RI APDU, this is an error and the association shall be released.

b) If the association is rejected, the Protocol-Version field contains the protocol versions of this Recommendation that can be supported.

c) If the association is accepted and if the Recovery-Context-Handle field is present, its value shall be stored in the recovery log (the log-ready record or the log-commit record, whichever applies, see 7.5) for all transactions on the association (except transactions on dialogues where a different Recovery-Context-Handle is received at dialogue establishment).

d) If the Functional-Unit-Capabilities field is present, subsequent TP-BEGIN-DIALOGUE-RI APDUs issued on this association shall not select functional units that are not identified in this field.

e) If the Functional-Unit-Capabilities field is absent, subsequent TP-BEGIN-DIALOGUE-RI APDUs issued on this association shall not select functional units that are not *basic functional units*.

The A-ASSOCIATE confirm with no embedded TP-INITIALIZE-RC APDU shall only be received when the association establishment was rejected by ACSE.

If the Application Context Name parameter on the A-ASSOCIATE confirm is different from the Application Context Name parameter on the A-ASSOCIATE request, the PM makes a local decision whether to:

1) accept the association using this new application context; or

2) release the association.

NOTE – When an association has been rejected, it is a local decision as to what to do next. The TPPM, based on a local decision, may notify the TPSUI that the dialogue is being rejected, it may try again (immediately) to establish the association, it may wait for some period of time and then retry, etc.

8.5.8 Initiating an association release

Release of an association may be initiated according to a local decision at any time the SACF is in the FREE state (see 10.2).

8.5.9 Aborting an association

An A-ABORT request may be issued by a U-ASE at any time. If a U-ASE issues an A-ABORT request, then the TPPM will take the actions appropriate to association abort which may include rolling back the current transaction or initiating recovery.

8.5.10 Initiating a dialogue solicitation

A TPPM may solicit the establishment of a dialogue on any association in the FREE state for which it is contention-loser and on which the Dialogue Solicitation functional unit is selected. A SAF-SOLICIT-DIALOGUE request shall be issued, with the parameters set as follows:

a) candidate-initiating-tpsu-titles shall list the TPSU titles at the peer AEI from which a dialogue establishment is solicited;

b) candidate-recipient-tpsu-titles shall list the TPSU titles at this AEI to which the dialogue establishment is solicited.

8.5.11 Responding to a dialogue solicitation

On receipt of an SAF-SOLICIT-DIALOGUE indication, it is a local decision whether to accept the solicitation.

Accepting a solicitation means initiating a dialogue establishment for which the initiating and responding TPSU titles match those proposed in the solicitation. The initiating TPSU titles are considered to match if either:

a) the candidate-initiating-tpsu-titles parameter on the SAF-SOLICIT-DIALOGUE indication is present and contains a list of one or more values, and the initiating-tpsu-title parameter on the TP-BEGIN-DIALOGUE request is one of those values; or

b) the candidate-initiating-tpsu-titles parameter on the SAF-SOLICIT-DIALOGUE indication is absent or is present but empty.

The initiating TPSU titles are considered to not match only if the candidate-initiating-tpsu-titles parameter on the SAF-SOLICIT-DIALOGUE indication is present and contains a list of one or more values but the initiating-tpsu-title parameter on the TP-BEGIN-DIALOGUE request is not one of those values.

The corresponding rule applies for the candidate-responding-tpsu-title parameter of the SAF-SOLICIT-DIALOGUE indication and the responding-tpsu-title parameter of the TP-BEGIN-DIALOGUE request.

If the decision is to accept the solicitation, local means shall be used to cause a TPSUI to issue a TP-BEGIN-DIALOGUE request with matching TPSU title parameters. When the TPPM is required to assign an association to the requested dialogue, it shall assign the association on which the SAF-SOLICIT-DIALOGUE indication was received.

NOTE – The means by which an appropriate TPSUI is created, if necessary, and stimulated to issue the TP-BEGIN-DIALOGUE request is a local matter.

If the association is already assigned to another dialogue or channel establishment (i.e. a dialogue request has already been issued in which the initiating and/or recipient TPSU titles parameters do not match the candidate lists on the SAF-SOLICIT-DIALOGUE indication), the solicitation is considered rejected, and there are no further actions as a result of the solicitation.

If the decision is to reject the solicitation and the association is not used for another dialogue or channel establishment, an SAF-SOLICIT-DIALOGUE response shall be issued.

8.5.12 Receiving a rejection of a dialogue solicitation

The rejection of a dialogue is signalled by the receipt of an SAF-SOLICIT-DIALOGUE confirm or the issue by the SACF of an AF-BEGIN-DIALOGUE indication in which the initiating TPSU title does not match those in the (non-empty) candidate-initiating-tpsu-titles parameter of the previous SAF-SOLICIT-DIALOGUE request or the recipient TPSU title does not match those in the (non-empty) candidate-recipient-tpsu-titles.

Following the rejection of a dialogue solicitation, it is a local decision what to do next.

NOTE – It is possible that a dialogue that matches the solicitation request has been established on a different association since the solicitation was initiated.

8.5.13 Successful dialogue solicitation

A successful dialogue solicitation is signalled by the SACF issuing an AF-BEGIN-DIALOGUE indication on which the initiating TPSU title matches one of those in the candidate-initiating-tpsu-titles parameter of the previous SAF-SOLICIT-DIALOGUE request and the recipient TPSU title matches one of those in the candidate-recipient-tpsu-titles of the request.

If the candidate parameter on the request was absent or empty, the corresponding parameter on the AF-BEGIN-DIALOGUE indication is considered to match, whatever its value (even if the parameter is absent on the indication).

The AF-BEGIN-DIALOGUE indication is treated as a normal dialogue establishment indication. The procedures and possible actions of the TPPM resulting from the issue of the AF-BEGIN-DIALOGUE and subsequent events on the association are not affected by the solicitation.

9 TP-ASE description

9.1 Introduction

This clause defines the Service Primitives provided by the TP-ASE to the MACF (through the applicable SACF procedures). It further defines the TP APDUs generated to, and received from, the partner TP-ASE. It also defines the TP APDU mappings onto CCR, ACSE, and Presentation services.

Table 19 lists the AF Service primitives together with associated TP APDUs.

9.2 AF Service Definition

The primitives and parameters of the AF services are given in 9.3. The sequence of primitives is given by the appropriate tables in 9.3, reading from left-to-right. This Recommendation (clauses 7 to 11 and, indirectly by reference, the TP Service) defines the constraints on, effects, and collisions of the service primitives.

9.2.1 AF-BEGIN-DIALOGUE

This service is initiated as a direct result of a TP-BEGIN-DIALOGUE request service primitive, or is initiated directly by the CPM to establish a channel.

This service is a confirmed service for channels and a conditionally confirmed service for dialogues.

NOTE – The only time this service is not confirmed is when the confirmation parameter of the AF-BEGIN-DIALOGUE request is set to "negative", the dialogue is not rejected, and a rollback initiating service primitive or a TPPM-initiated rollback occurs at the partner prior to any other requests or responses by that partner. In this case, a C-ROLLBACK indication or confirm will serve to confirm the dialogue establishment.

The AF-BEGIN-DIALOGUE response and confirm are used as delimiters for discarding stray APDUs.

When the response and confirm service primitives are referenced in the procedure clauses, the first value listed in the parenthetical argument is the value of the Result parameter; the second is the value of the Mapping parameter.

Table 19/X.862 – AF Service Primitives and Associated TP APDUs

AF Service Primitives	TP APDUs
AF-BEGIN-DIALOGUE req/ind AF-BEGIN-DIALOGUE rsp/cnf	TP-BEGIN-DIALOGUE-RI TP-BEGIN-DIALOGUE-RC
AF-BID req/ind AF-BID rsp/cnf	TP-BID-RI TP-BID-RC
AF-END-DIALOGUE req/ind AF-END-DIALOGUE rsp/cnf	TP-END-DIALOGUE-RI TP-END-DIALOGUE-RC
AF-U-ERROR req/ind AF-U-ERROR rsp/cnf	TP-U-ERROR-RI TP-U-ERROR-RC
AF-ABORT req/ind	TP-ABORT-RI
AF-ABORT-AND-REPORT req/ind	TP-ABORT-RI + TP-REPORT-RI or TP-ABORT-AND-REPORT-RI
AF-GRANT-CONTROL req/ind	TP-GRANT-CONTROL-RI
AF-REQUEST-CONTROL req/ind	TP-REQUEST-CONTROL-RI
AF-HANDSHAKE req/ind AF-HANDSHAKE rsp/cnf	TP-HANDSHAKE-RI TP-HANDSHAKE-RC
AF-HANDSHAKE-AND-GRANT-CONTROL req/ind AF-HANDSHAKE-AND-GRANT-CONTROL rsp/cnf	TP-HANDSHAKE-AND-GRANT-CONTROL-RI TP-HANDSHAKE-AND-GRANT-CONTROL-RC
AF-DEFER req/ind	TP-DEFER-RI
AF-BEGIN-TRANSACTION req/ind	TP-BEGIN-TRANSACTION-RI
AF-PREPARE req/ind	TP-PREPARE-RI
AF-NOCHANGE req/ind	TP-NEXT-TID-RI
AF-REPORT req/ind	TP-HEURISTIC-REPORT-RI
AF-TOKEN-GIVE req/ind	TP-TOKEN-GIVE-RI
AF-TOKEN-PLEASE req/ind	TP-TOKEN-PLEASE-RI
AF-RECOVER req/ind	TP-RECOVER-RI
AF-SOLICIT-DIALOGUE req/ind AF-SOLICIT-DIALOGUE rsp/cnf	TP-SOLICIT-DIALOGUE-RI TP-SOLICIT-DIALOGUE-RC

9.2.2 AF-BID

This confirmed service is used by the SACF in order to gain the rights of the contention-winner temporarily for attempting establishment of the subsequent dialogue or channel.

When the response and confirm service primitives are referenced in the procedure clauses, the value listed in the parenthetical argument is the value of the Result parameter.

9.2.3 AF-END-DIALOGUE

This service is initiated as a direct result of a TP-END-DIALOGUE request service primitive, or is initiated directly by the CPM to terminate a channel.

This is an optionally confirmed service for dialogues. This is an unconfirmed service for channels.

9.2.4 AF-U-ERROR

This service is initiated as a direct result of a TP-U-ERROR request service primitive.

This is an unconfirmed service in Polarized Control.

In Shared Control, this is an unconfirmed service when it serves as a negative response to a prior AF-HANDSHAKE indication or an AF-END-DIALOGUE indication with the Confirmation parameter set to "true". Otherwise, this is a confirmed service and the request primitive begins the *user error purging period*; in this case, either an AF-U-ERROR confirm, an AF-HANDSHAKE indication, or an AF-END-DIALOGUE indication with the Confirmation parameter set to "true" serves as the confirmation to the AF-U-ERROR request (but see the definition of *user error purging period* in 7.3.53 for more details when more than one AF-U-ERROR request is unconfirmed).

9.2.5 AF-ABORT

This service is initiated as a direct result of a TP-U-ABORT request service primitive, or is initiated directly by the PM to abnormally terminate a dialogue or channel.

This is an unconfirmed service.

When this service primitive is referenced in the procedure clauses, the first value listed in the parenthetical argument is the value of the Type parameter; the second is the value of the Mapping parameter.

9.2.6 AF-GRANT-CONTROL

This service is initiated as a direct result of a TP-GRANT-CONTROL request service primitive.

This is an unconfirmed service.

9.2.7 AF-REQUEST-CONTROL

This service is initiated as a direct result of a TP-REQUEST-CONTROL request service primitive.

This is an unconfirmed service.

9.2.8 AF-HANDSHAKE

This service is initiated as a direct result of a TP-HANDSHAKE request service primitive.

This is a confirmed service.

9.2.9 AF-HANDSHAKE-AND-GRANT-CONTROL

This service is initiated as a direct result of a TP-HANDSHAKE-AND-GRANT-CONTROL request service primitive.

This service is a confirmed service.

9.2.10 AF-DEFER

This service is initiated as a result of either a TP-DEFERRED-END-DIALOGUE or a TP-DEFERRED-GRANT-CONTROL request service primitive.

This is an unconfirmed service.

When this service primitive is referenced in the procedure clauses, the value listed in the parenthetical argument is the value of the Type parameter.

9.2.11 AF-BEGIN-TRANSACTION

This service is initiated as a direct result of a TP-BEGIN-TRANSACTION request service primitive when the checking of ready directions is not required in the subtree.

This is an unconfirmed service.

9.2.12 AF-PREPARE

This service is initiated as a direct result of either a TP-PREPARE request service primitive or a TP-COMMIT request service primitive by a superior.

This is an unconfirmed service.

When this service primitive is referenced in the procedure clauses, the value listed in the parenthetical argument is the value of the Data-permitted parameter, with "true" represented as "data-permitted" and "false" represented as "no-data-permitted".

9.2.13 AF-REPORT

This service is initiated by a subordinate as a result of the existence of log-damage in the subtree, or, when the subordinate sent the commit request to the superior, to report that there is no log-damage in the subtree.

This service is also used when the Completion Diagnostics functional unit is selected. It is initiated by a subordinate to send Completion-data, Diagnostic and Severity information to the superior.

This is an unconfirmed service.

When this service primitive is referenced in the procedure clauses, the value listed in the parenthetical argument is the value of the Mapping parameter.

9.2.14 AF-ABORT-AND-REPORT

This service is initiated by a subordinate as a result of the existence of log-damage in the subtree along with a dialogue abort condition. If the Completion Diagnostics functional unit is selected, this service is initiated by a subordinate to send Completion-data and Severity information to the superior.

This is a combination of the AF-ABORT and AF-REPORT services and has the combined semantics of the AF-ABORT and AF-REPORT services.

This is an unconfirmed service.

When this service primitive is referenced in the procedure clauses, the value listed in the parenthetical argument is the value of the Mapping parameter.

9.2.15 AF-NOCHANGE

The request and indication of this service is initiated by a superior when initiating one-phase commitment on a dialogue with the Chained Transactions functional unit selected.

It is an indirectly-confirmed service. A C-NOCHANGE response and confirm will confirm the AF-NOCHANGE service.

9.2.16 AF-EARLY-EXIT

This service is initiated as a direct result of a TP-EARLY-EXIT request service primitive.

This is a confirmed service.

9.2.17 AF-RECOVER

This service enables TPPMs to invoke recovery after a failure when the recovery-context-handle is provided.

This service is unconfirmed.

When this service primitive is referenced in the procedure clauses, the value listed in the parenthetical argument is the value of the Recovery-State parameter.

9.2.18 AF-TOKEN-GIVE

This service is used to transfer the *token* to the peer PM either during two-way-recovery or while establishing or terminating a dialogue.

This is an unconfirmed service.

9.2.19 AF-TOKEN-PLEASE

This service is used to request the *token* from the peer PM; it will be used by CPMs only.

This is an unconfirmed service.

9.2.20 AF-SOLICIT-DIALOGUE

This service is initiated to solicit the establishment of a dialogue. The service is initiated as part of the management of associations.

This is an optionally confirmed service. It is confirmed only when the solicitation is explicitly rejected.

When this service is referenced in the procedure clauses, the value listed in the parenthetical argument is the value of the Mapping parameter.

9.3 AF-Services and TP APDUs: Parameters and field Mappings

9.3.1 AF-BEGIN-DIALOGUE request/indication/response/confirm, TP-BEGIN-DIALOGUE-RI/-RC APDU

Tables 20 and 21 give the parameters and field mappings for Dialogue Establishment and Channel Establishment, respectively. The parameters and fields are as described in the "Primitives and parameters" subclause for the TP-BEGIN-DIALOGUE service in ITU-T Rec. X.861 | ISO/IEC 10026-2, with the following exceptions:

a) **Functional-Units**: The meaning and usage of this parameter/field is as described for the TP-BEGIN-DIALOGUE Service in ITU-T Rec. X.861 | ISO/IEC 10026-2, except that when it is used on a dialogue, the Cancel Functional Unit is optional and when it is used on a Channel, the parameter specifies only the Recovery Functional Unit (see 13.1.3.4).

b) **Result**: The meaning and usage of this parameter/field is as described for the TP-BEGIN-DIALOGUE Service in ITU-T Rec. X.861 | ISO/IEC 10026-2, except that the value "rejected(provider)" is permitted on the AF-BEGIN-DIALOGUE response. For a channel, the value "rejected(user)" is not used.

c) **Diagnostic**: The meaning and usage of this parameter/field is as described for the TP-BEGIN-DIALOGUE Service in ITU-T Rec. X.861 | ISO/IEC 10026-2, except that the following additional values are allowed:

   1) "two-way-recovery-not-supported" when "two-way-recovery" was the value of the Channel-Utilization parameter/field. This value only applies to channels;

   2) "association-reserved" when the contention-winner has reserved the association for its use;

   3) "tppm-recovery-not-available" when the CPM is unable to support recovery, due to a local condition. This value only applies to channels.

   The following values are not allowed:

   1) "ready flow check fails";

   2) "may-send-ready-settings-rejected".

   The following values do not apply to channels:

   1) "recipient-tpsu-title-unknown";

   2) "tpsu-not-available(permanent)";

   3) "tpsu-not-available(transient)";

   4) "recipient-tpsu-title-required";

   5) "functional-unit-combination-not-supported";

   6) "subordinate-must be commit-superior";

   7) "subordinate-must-be-commit-subordinate".

d) **Correlator**: A correlator unambiguous within the scope of a single direction of transfer in the SAO.

   Due to the use of unconfirmed services and re-use of associations, it may happen that APDUs that are foreign to a particular dialogue are received. Such APDUs are referred to as "stray APDUs".

   To resolve such an ambiguity, a correlator is conveyed with the request for, and the acknowledgement/rejection of, a dialogue/channel establishment to identify the dialogue/channel.

   NOTE – A correlator must be unique within the scope of the correlators used by the SACF on its previous issued and unacknowledged dialogue/channel establishment requests. Once a dialogue/channel establishment request is acknowledged (by acceptance or rejection), all unacknowledged correlators used prior to that returned are acknowledged and available for re-use.

   The value of this parameter is set on the request/response when the service primitive passes through the SACF.

e) **Recovery-context-handle**: The use of this parameter is as described for the corresponding parameter on the TP-INITIALIZE service, but shall apply only to the log-records of transactions on the dialogue being established. The parameter shall only be present if, when the supporting association was established, on whichever of TP-INITIALIZE-RI and TP-INITIALIZE-RC APDUs was received, the RCH-on-dialogue functional unit was included in the Functional-Unit-Capabilities, and on whichever of the TP-INITIALIZE-RI and TP-INITIALIZE-RC APDUs was sent, the Senders-RCH-Varies field had the value "true".

f) **Channel-Utilization**: Identifies the use of the channel. It takes one of the following values:

   1) "one-way-recovery";

   2) "two-way-recovery";

   depending on the desired mode of recovery and is present only if the Recovery Functional Unit (see 13.1.3.4) is selected in the value of the Functional-Units parameter of the AF-BEGIN-DIALOGUE request;

g) **Mapping**: Indicates the underlying service to which this AF service is mapped. For AF-BEGIN-DIALOGUE response, it may have one of the following values:

   1) "dataRI" – Mapped to P-DATA request;

   2) "rollbackRI" – Mapped to C-ROLLBACK request;

   3) "rollbackRC" – Mapped to C-ROLLBACK response.

   For AF-BEGIN-DIALOGUE confirm, it may have one of the following values:

   1) "dataRI" – Mapped from P-DATA indication;

   2) "rollbackRI" – Mapped from C-ROLLBACK indication;

   3) "rollbackRC" – Mapped from C-ROLLBACK confirm.

h) **Last-Partner-Identifier**: Contains the value of the Correlator of the last TP-BEGIN-DIALOGUE-RI APDU received by the contention-loser. If no dialogue has been established on the association, the bid mechanism has been used for this dialogue, or the requestor is the contention-winner, this parameter/field does not exist.

9.3.2 AF-BID request/indication/response/confirm, TP-BID-RI/-RC APDU

Table 22 gives the parameters and field mappings for the bid mechanism. The parameters and fields are as follows:

a) **CCR-Token-Requested**: Indicates whether the *token* required by CCR is requested. It takes the following values:

   1) "true", when the *token* is requested;

   2) "false", when the *token* is not requested.

b) **Last-Partner-Identifier**: Contains the value of the Correlator of the last TP-BEGIN-DIALOGUE-RI APDU received by the contention-loser. If no dialogue has been established on the association, this parameter/field does not exist.

c) **Result**: Indicates whether the contention-loser has been temporarily granted the rights of the contention-winner. It takes the following values:

   1) "accepted", when the rights are granted;

   2) "rejected", when the rights are not granted.

9.3.3 AF-END-DIALOGUE request/indication/response/confirm – TP-END-DIALOGUE-RI/-RC APDU

Table 23 gives the parameter and field mappings for Dialogue Termination. The parameter and field are as described in the "Primitives and parameters" subclause for the TP-END-DIALOGUE service in ITU-T Rec. X.861 | ISO/IEC 10026-2.

Table 24 gives the parameter and field mappings for Channel Termination. The parameter and field are as follows:

– **Confirmation**: This must be set to "false" (see 11.4.6, "Terminating a channel").

The AF-END-DIALOGUE response and confirm primitives have no parameters; the TP-END-DIALOGUE-RC APDU has no fields.

Table 20/X.862 – Dialogue Establishment mappings

Service Primitive/TP APDU	TP-	AF-	TP-	AF-	TP-		TP-	AF-	TP-	AF-	TP-
	----------- BEGIN-DIALOGUE ------------										
Parameter/Field	**req**	**req**	**-RI**	**ind**	**ind**		**rsp**	**rsp**	**-RC**	**cnf**	**cnf**
Initiating-AP-Title					O						
Initiating-API-Identifier					O						
Initiating-AE-Qualifier					O						
Initiating-AEI-Identifier					O						
Initiating-TPSU-Title	U	(=)	(=)	(=)	(=)						
Recipient-AP-Title	M										
Recipient-API-Identifier	C										
Recipient-AE-Qualifier	C										
Recipient-AEI-Identifier	C										
Recipient-TPSU-Title	U	(=)	(=)	(=)							
Functional-Units	M	(=)	(=)	(=)	(=)			C	(=)	(=)	(=)
Quality-of-Service	U										
Application Context Name	M										
Begin-Transaction	C	(=)	(=)	(=)	(=)						
Confirmation	M	(=)	(=)	(=)	(=)						
Result							M	(=)/M	(=)	(=)	(=)
Diagnostic								C	(=)	(=)/M	(=)
Rollback											M
Correlator		M	(=)	(=)				(=)	(=)	(=)	
Channel-Utilization											
Mapping		M		(=)				M		(=)	
Last-Partner-Identifier		C	(=)	(=)							
Superior-may-send-ready	C	(=)	(=)	(=)	(=)						
Subordinate-may-send-ready	C	(=)	(=)	(=)	(=)						
Check-ready-directions	C	(=)	(=)	(=)	(=)						
Recovery-context-handle		C	(=)	(=)				C	(=)	(=)	
User-Data	U	(=)	(=)	(=)	(=)		U	(=)	(=)	(=)	(=)

Table 21/X.862 – Channel Establishment mappings

Service Primitive/TP APDU	AF-	TP-	AF-		AF-	TP-	AF-
	------------ BEGIN-DIALOGUE ------------						
Parameter/Field	**req**	**-RI**	**ind**		**rsp**	**-RC**	**cnf**
Initiating-AP-Title							
Initiating-API-Identifier							
Initiating-AE-Qualifier							
Initiating-AEI-Identifier							
Initiating-TPSU-Title							
Recipient-AP-Title							
Recipient-API-Identifier							
Recipient-AE-Qualifier							
Recipient-AEI-Identifier							
Recipient-TPSU-Title							
Functional-Units	M	(=)	(=)				
Quality-of-Service							
Application Context Name							
Begin-Transaction							
Confirmation							
Result					M	(=)	(=)
Diagnostic					C	(=)	(=)
Rollback							
Correlator	M	(=)	(=)		(=)	(=)	(=)
Channel-Utilization	M	(=)	(=)				
Mapping					M		(=)
Last-Partner-Identifier	C	(=)	(=)				
User-Data							

Table 22/X.862 – Bid mechanism mappings

Service Primitive/TP APDU	AF-	TP-	AF-		AF-	TP-	AF-
	------------ BID ------------						
Parameter/Field	**req**	**-RI**	**ind**		**rsp**	**-RC**	**cnf**
CCR-Token-Requested	M	(=)	(=)				
Last-Partner-Identifier	C	(=)	(=)				
Result					M	(=)	(=)

Table 23/X.862 – Dialogue termination mappings

Service Primitive/TP APDU	TP-	AF-	TP-	AF-	TP-
	----- END-DIALOGUE -----				
Parameter/Field	**req**	**req**	**RI**	**ind**	**ind**
Confirmation	M	(=)	(=)	(=)	(=)

Table 24/X.862 – Channel termination mappings

Service Primitive/TP APDU	AF-	TP-	AF-
	-- END-DIALOGUE --		
Parameter/Field	**req**	**RI**	**ind**
Confirmation	M	(=)	(=)
NOTE – The confirmation parameter is always "false" for Channel Termination (see 11.4.6, "Terminating a channel").			

9.3.4 AF-U-ERROR request/indication/response/confirm – TP-U-ERROR-RI/-RC APDU

These services have no parameters and these TP APDUs have no fields.

9.3.5 AF-ABORT request/indication – TP-ABORT-RI APDU

Tables 25 and 26 give the parameters and field mappings for User Abort and Provider Abort, respectively. The parameters and fields are as described in the "Primitives and parameters" subclause for the TP-U-ABORT and TP-P-ABORT services in ITU-T Rec. X.861 | ISO/IEC 10026-2, with the following exceptions:

a) **Type**: Indicates the type of abort (user or provider). It takes one of the following values:

   1) "user";

   2) "provider".

b) **Mapping**: Indicates the underlying service to which this AF service is mapped. For AF-ABORT request, it takes one of the following values:

   1) "abortRI" – Mapped to A-ABORT request;

   2) "dataRI" – Mapped to P-DATA request;

   3) "commitRI" – Mapped to C-COMMIT request;

   4) "commitRC" – Mapped to C-COMMIT response;

   5) "rollbackRI" – Mapped to C-ROLLBACK request;

   6) "rollbackRC" – Mapped to C-ROLLBACK response;

   7) "nochangeRC" – Mapped to C-NOCHANGE response.

   For AF-ABORT indication, it takes one of the following values:

   1) "abortRI" – Mapped from A-ABORT indication;

   2) "dataRI" – Mapped from P-DATA indication;

   3) "commitRI" – Mapped from C-COMMIT indication;

   4) "commitRC" – Mapped from C-COMMIT confirm;

   5) "rollbackRI" – Mapped from C-ROLLBACK indication;

   6) "rollbackRC" – Mapped from C-ROLLBACK confirm;

   7) "nochangeRC" – Mapped from C-NOCHANGE confirm.

c) **Diagnostic**: Does not take the values "end-dialogue-collision" or "begin-transaction-end-dialogue-collision".

NOTE – These values are generated locally by the MACF.

d) **Outcome**: Only used if mapping parameter has the value "nochangeRC". Takes the values "commit" or "not-determined".

Table 25/X.862 – User Abort mappings

Service Primitive/TP APDU	**TP-U**	**AF-**	**TP-**	**AF-**	**TP-U**
	------------ ABORT ------------				
Parameter/Field	**req**	**req**	**RI**	**ind**	**ind**
Type		M	(=)	(=)	
Mapping		M		(=)	
Diagnostic					
Rollback					M
User-Data	U	(=)	(=)	(=)	(=)

Table 26/X.862 – Provider Abort mappings

Service Primitive/TP APDU	**AF-**	**TP-**	**AF-**	**TP-P**
	------------ ABORT ------------			
Parameter/Field	**req**	**RI**	**ind**	**ind**
Type	M	(=)	(=)	
Mapping	M		(=)	
Diagnostic	M	(=)	(=)	(=)/M
Rollback				M
Outcome	C			(=)
User-Data				

9.3.6 AF-GRANT-CONTROL request/indication – TP-GRANT-CONTROL-RI APDU

These services have no parameters and this TP APDU has no fields.

9.3.7 AF-REQUEST-CONTROL request/indication – TP-REQUEST-CONTROL-RI APDU

These services have no parameters and this TP APDU has no fields.

9.3.8 AF-HANDSHAKE request/indication/response/confirm – TP-HANDSHAKE-RI/-RC APDU

Table 27 gives the parameter and field mappings for Handshake. The parameter and field are as described in the "Primitives and parameters" subclause for the TP-HANDSHAKE service in ITU-T Rec. X.861 | ISO/IEC 10026-2.

The AF-HANDSHAKE response and confirm primitives have no parameters; the TP-HANDSHAKE-RC APDU has no fields.

9.3.9 AF-HANDSHAKE-AND-GRANT-CONTROL request/indication/response/confirm – TP-HANDSHAKE-AND-GRANT-CONTROL-RI/-RC APDU

Table 28 gives the parameter and field mappings for Handshake and Grant Control. The parameter and field are as described in the "Primitives and parameters" subclause for the TP-HANDSHAKE-AND-GRANT-CONTROL service in ITU-T Rec. X.861 | ISO/IEC 10026-2.

The AF-HANDSHAKE-AND-GRANT-CONTROL response and confirm primitives have no parameters; the TP-HANDSHAKE-AND-GRANT-CONTROL-RC APDU has no fields.

Table 27/X.862 – Handshake mappings

Service Primitive/TP APDU	TP-	AF-	TP-	AF-	TP-
	--------- HANDSHAKE ---------				
Parameter/Field	**req**	**req**	**RI**	**ind**	**ind**
Confirmation-Urgency	C	(=)	(=)		

Table 28/X.862 – Handshake and Grant Control mappings

Service Primitive/TP APDU	TP-	AF-	TP-	AF-	TP-
	HANDSHAKE-AND-GRANT-CONTROL				
Parameter/Field	**req**	**req**	**RI**	**ind**	**ind**
Confirmation-Urgency	M	(=)	(=)		

9.3.10 AF-BEGIN-TRANSACTION request/indication – TP-BEGIN-TRANSACTION-RI APDU

Table 29 gives the parameter and field mappings for TP/AF-BEGIN-TRANSACTION. The parameters and fields are as described in the "C-BEGIN parameters" subclause in ITU-T Rec. X.851 | ISO/IEC 9804, with the following addition:

– **Check-ready-directions**: The meaning and usage of this field is as described in the "Primitives and parameters" subclause for the TP-BEGIN-TRANSACTION service in ITU-T Rec. X.861 | ISO/IEC 10026-2.

If the Check-ready-directions parameter on the TP-BEGIN-TRANSACTION request has the default value "true", this AF-Service and this TP-ADPU shall not be used.

Table 29/X.862 – TP/AF-BEGIN-TRANSACTION mappings

Service Primitive/TP APDU	TP-	AF-	TP-	AF-	TP-
	---- BEGIN-TRANSACTION ----				
Parameter/Field	**req**	**req**	**RI**	**ind**	**ind**
Check-ready-directions	C	(=)	(=)	(=)	(=)
Atomic Action Identifier		M		(=)	
Branch Identifier		M		(=)	

9.3.11 AF-DEFER request/indication – TP-DEFER-RI APDU

Table 30 gives the parameter and field mappings for Deferred End Dialogue and Deferred Grant Control. The parameter and field are as described as follows:

– **Type**: Indicates the type of Deferred service (End Dialogue or Grant Control). It takes one of the following values:

1) "end-dialogue";

2) "grant-control".

Table 30/X.862 – Deferred End Dialogue and Deferred Grant Control mappings

Service Primitive/TP APDU	TP- D-E-D or D-G-C[a)]	AF-	TP-	AF-
		---------- DEFER ----------		
Parameter/Field	**req**	**req**	**RI**	**ind**
Type		M	(=)	(=)
[a)] TP-DEFERRED-END-DIALOGUE or TP-DEFERRED-GRANT-CONTROL				

9.3.12 AF-PREPARE request/indication – TP-PREPARE-RI APDU

Table 31 gives the parameter and field mappings for TP/AF-PREPARE. The parameters and fields are as described in the "Primitives and parameters" subclauses for TP-PREPARE request and TP-PREPARE INDICATION in ITU-T Rec. X.861 | ISO/IEC 10026-2.

Table 31/X.862 – TP/AF-PREPARE mappings

Service Primitive/TP APDU	TP-	AF-	TP-	AF-	TP-
	----------- PREPARE -----------				
Parameter/Field	**req**	**req**	**RI**	**ind**	**ind**
Data-Permitted	C	(=)	(=)	(=)	(=)

9.3.13 AF-REPORT request/indication – TP-REPORT-RI APDU

Table 32 gives the parameters and field mappings for Heuristic and Completion Reporting. The parameters and fields are as follows:

a) **Mapping**: Indicates the underlying service to which this AF service is mapped. For AF-REPORT request, it takes one of the following values:

   1) "commitRC" – Mapped to C-COMMIT response;

   2) "recoverCommitRI" – Mapped to C-RECOVER (commit) request;

   3) "recoverDoneRC" – Mapped to C-RECOVER (done) response;

   4) "rollbackRI" – Mapped to C-ROLLBACK request;

   5) "rollbackRC" – Mapped to C-ROLLBACK response;

   6) "dataRI" – Mapped to P-DATA request.

   For AF-REPORT indication, it takes one of the following values:

   1) "commitRC" – Mapped from C-COMMIT confirm;

   2) "recoverCommitRI" – Mapped from C-RECOVER (commit) confirm;

   3) "recoverDoneRC" – Mapped from C-RECOVER (done) confirm;

   4) "rollbackRI" – Mapped from C-ROLLBACK indication;

   5) "rollbackRC" – Mapped from C-ROLLBACK confirm;

   6) "dataRI" – Mapped from P-DATA indication.

b) **Heuristic-report**: This parameter is as described the "Primitives and parameters" subclause for the TP-HEURISTIC-REPORT subclause in ITU-T Rec. X.861 | ISO/IEC 10026-2. Its presence is as described in the definition of "*carrying the reporting status*".

c) **Severity**, **Diagnostic**: These parameters are as described the "Primitives and parameters" subclause for the TP-COMPLETION-REPORT subclause in ITU-T Rec. X.861 | ISO/IEC 10026-2. Their presence is as described in the definition of "*carrying the reporting status*".

d) **Completion-Data**: The presence of this parameter is as described in the definition of "*carrying the reporting status*". On the AF-REPORT request, the parameter contains the value from the Completion-Data parameter of the most recent TP-DONE request on which that parameter was present, if any TP-DONE request had that parameter present. If no TP-DONE request had the Completion-Data parameter present, the Completion-Data parameter is absent from the AF-ABORT request. The Completion-Data parameter of the AF-REPORT indication, if present, is used for the Completion-Data parameter of the TP-COMPLETION-REPORT indication.

e) **Atomic Action Identifier** and **Branch Identifier**: These parameters are as described in the "C-RECOVER parameters" subclause in ITU-T Rec. X.851 | ISO/IEC 9804.

f) **Recovery-Context-Handle**: This parameter is absent unless the Mapping parameter has the value "recoverCommitRI". Takes the value of the field with the same name received on the TP-BEGIN-DIALOGUE-RI/RC APDU, if the parameter was present, or on the TP-INITIALIZE-RI/RC APDU from the partner AEI if the parameter was not present on the received TP-BEGIN-DIALOGUE-RI/RC APDU. If a Recovery-Context-Handle does not exist for the remote partner, then the parameter is absent on the AF-REPORT request.

Table 32/X.862 – Heuristic Reporting mappings

Service Primitive/TP APDU	**AF-**	**TP-**	**AF-**	**TP-HEURISTIC-REPORT**	**TP-COMPLETION-REPORT**
	------ REPORT ------				
Parameter/Field	**req**	**RI**	**ind**	**ind**	**ind**
Mapping	M		(=)		
Heuristic-Report	C[a)]	(=)	(=)	(=)	
Severity	C[a)]	(=)	(=)		(=)
Diagnostic	C[a)]	(=)	(=)		(=)
Completion-Data	C[a)]	(=)	(=)		(=)
Atomic Action Identifier	C[b)]		C[c)]		
Branch Identifier	C[b)]		C[c)]		
Recovery-Context-Handle	C[d)]		(=)		

[a)] The presence and, if present, the values of these parameters are specified in the definition *carrying the reporting status*.

[b)] These parameters exist if the Mapping parameter is recoverCommitRI or recoverDoneRC and the service is on a channel.

[c)] These parameters exist if the Mapping parameter is recoverCommitRI and is then the same as the value on the AF-REPORT request.

[d)] This parameter may exist only if the Mapping parameter is recoverCommitRI.

9.3.14 AF-ABORT-AND-REPORT request/indication

Table 33 gives the parameters mappings for the combined Auxiliary Facility Service AF-ABORT-AND-REPORT. The parameters are as follows:

a) **Mapping**: Indicates the underlying service to which this AF service is mapped. For AF-ABORT-AND-REPORT request, it takes one of the following values:

1) "commitRC" – Mapped to C-COMMIT response;

2) "rollbackRI" – Mapped to C-ROLLBACK request;

3) "rollbackRC" – Mapped to C-ROLLBACK response;

4) "dataRI" – Mapped to P-DATA request.

For AF-ABORT-AND-REPORT indication, it takes one of the following values:

1) "commitRC" – Mapped from C-COMMIT confirm;
2) "rollbackRI" – Mapped from C-ROLLBACK indication;
3) "rollbackRC" – Mapped from C-ROLLBACK confirm;
4) "dataRI" – Mapped from P-DATA indication.

The TP-ABORT-AND-REPORT-RI APDU is only used if the Mapping parameter has the value "dataRI". In all other cases, the TP-ABORT-RI and TP-REPORT-RI APDUs are used.

b) **Heuristic-report**: This parameter is as described in the "Primitives and parameters" subclause for the TP-HEURISTIC-REPORT subclause in ITU-T Rec. X.861 | ISO/IEC 10026-2. Its presence is as described in the definition of "*carrying the reporting status*".

c) **User-Data**: This parameter is as described in the TP-U-ABORT subclause of ITU-T Rec. X.861 | ISO/IEC 10026-2. The parameter is present on the AF-ABORT-AND-REPORT service if the TP-U-ABORT request has a User-Data parameter.

d) **Severity**, **Diagnostic**: These parameters are as described in the "Primitives and parameters" subclause for the TP-COMPLETION-REPORT subclause in ITU-T Rec. X.861 | ISO/IEC 10026-2. Their presence is as described in the definition of "*carrying the reporting status*".

e) **Completion-Data:** The parameter of the AF-ABORT-AND-REPORT request contains the value from the Completion-Data parameter of the most recent TP-DONE request on which that parameter was present, if any TP-DONE request had that parameter present. If no TP-DONE request had the Completion-Data parameter present, the Completion-Data parameter is absent from the AF-ABORT-AND-REPORT request. The Completion-Data parameter of the AF-ABORT-AND-REPORT indication, if present, is used for the Completion-Data parameter of the TP-COMPLETION-REPORT indication.

Table 33/X.862 – AF-ABORT-AND-REPORT request/indication

Service Primitive/TP APDU	**AF-**	**TP-**	**AF-**
	ABORT-AND-REPORT		
Parameter/Field	**req**	**-RI**	**ind**
Mapping	M		(=)
Heuristic-Report	C	(=)	(=)
User-Data	C	(=)	(=)
Severity	C	(=)	(=)
Diagnostic	C	(=)	(=)
Completion-Data	C	(=)	(=)

9.3.15 AF-NOCHANGE request/indication – TP-NEXT-TID-RI APDU

Table 34 gives the parameter and field mappings for TP-ONE-PHASE, AF-NOCHANGE. The syntax of the first two parameters and fields is as described in the "C-BEGIN parameters" subclause in ITU-T Rec. X.851 | ISO/IEC 9804.

NOTE – The TP APDU (used only with Static One-phase dialogues) is named for the information it carries rather than the service that causes it to be sent.

Table 34/X.862 – TP-ONE-PHASE and AF-NOCHANGE mappings

Service Primitive/TP APDU	**TP-** **ONE-PHASE**	**AF-** **NOCHANGE**	**TP-** **NEXT-TID**	**AF-** **NOCHANGE**	**TP-** **ONE-PHASE**
Parameter/Field	**req**	**req**	**-RI**	**ind**	**ind**
Transaction-Identifier		M	(=)	(=)	
Branch-Suffix		M	(=)	(=)	

9.3.16 AF-EARLY-EXIT request/indication/response/confirm – TP-EARLY-EXIT-RI/-RC APDU

Table 35 gives the parameter and field mappings for Early-exit. The parameters and fields for the TP-EARLY-EXIT-RI are as described in the "Primitives and parameters" subclause for the TP-EARLY-EXIT request service in ITU-T Rec. X.861 | ISO/IEC 10026-2.

The AF-EARLY-EXIT response and confirm primitives have no parameters; the TP-EARLY-EXIT-RC APDU has no fields.

Table 35/X.862 – Early-exit mappings

Service Primitive/TP APDU	TP-	AF-	TP-	AF-	TP-
	------------ EARLY-EXIT ------------				
Parameter/Field	**req**	**req**	**RI**	**ind**	**ind**
Severity	C	(=)	(=)	(=)	(=)
Completion-Data	U	(=)	(=)	(=)	(=)

9.3.17 AF-RECOVER request/indication – TP-RECOVER-RI APDU

Table 36 gives the parameters and field mappings for TP/AF-RECOVER. The parameters and fields are as described in the "C-RECOVER parameters" subclause in ITU-T Rec. X.851 | ISO/IEC 9804, with the following exception:

– **Recovery-Context-Handle**: Takes the value of the field with the same name received on the TP-BEGIN-DIALOGUE-RI/RC APDU, if the parameter was present, or on the TP-INITIALIZE-RI/RC APDU from the partner AEI if the parameter was not present on the received TP-BEGIN-DIALOGUE-RI/RC APDU.

If a Recovery-Context-Handle does not exist for the remote partner, then this AF-Service and this TP APDU shall not be used.

Table 36/X.862 – TP/AF-RECOVER mappings

Service Primitive/TP APDU	AF-	TP-	AF-
	------ RECOVER ------		
Parameter/Field	**req**	**RI**	**ind**
Recovery-State	M		(=)
Recovery-Context-Handle	M	(=)	(=)
Atomic Action Identifier	M		(=)
Branch Identifier	M		(=)

9.3.18 AF-TOKEN-GIVE request/indication – TP-TOKEN-GIVE-RI APDU

Table 37 gives the parameters and field mappings for the AF-TOKEN-GIVE service. The parameters and fields are as described following:

a) **Reason**: Indicates the reason for which the *token* is transferred. It takes one of the following values:

1) regular – The *token* is transferred under the following conditions:

i) when it arrives at the contention-loser outside of a dialogue (that is, in the SACF FREE, STRAY, SOLICITING or BIDDING states);

ii) when it arrives at the contention-winner after an AF-BID (accepted) response request is issued;

iii) after the issuance of a *rollback response* or the receipt of a *rollback confirm* outside of a dialogue (that is, in the SACF CLEANUP ROLLBACK INDICATION EXPECTED or CLEANUP ROLLBACK CONFIRM EXPECTED states) by the contention-loser.

2) keep – The *token* is transferred under the following conditions:

   i) when an AF-BEGIN-DIALOGUE indication is received by the contention-winner and the bid mechanism was not used;

   ii) when an AF-TOKEN-GIVE (regular) indication is received by the contention-winner within a dialogue (that is, in the SACF BUSY state).

3) two-way-recovery – The *token* is transferred on a two-way-recovery channel after the issuance of a C-RECOVER request or AF-RECOVER request.

Table 37/X.862 – TP/AF-TOKEN-GIVE mappings

Service Primitive/TP APDU	**AF-**	**TP-**	**AF-**
	----- TOKEN-GIVE -----		
Parameter/Field	**req**	**RI**	**ind**
Reason	M	(=)	(=)
Correlator	C	(=)	(=)

b) **Correlator**: Contains the value of the Correlator of the last TP-BEGIN-DIALOGUE-RI APDU received by the contention-winner. The Correlator of the AF-TOKEN-GIVE request is present if the value of the Reason parameter is set to "keep".

9.3.19 AF-TOKEN-PLEASE request/indication – TP-TOKEN-PLEASE-RI APDU

These services have no parameters and this TP APDU has no fields.

9.3.20 AF-SOLICIT-DIALOGUE request/indication/response/confirm

Table 38 gives the parameters of this service:

Table 38/X.862 – Dialogue solicitation mappings

Service Primitive/TP APDU	**SAF**	**AF**	**TP**	**AF**	**SAF**		**SAF**	**AF**	**TP**	**AF**	**SAF**
	------------ SOLICIT-DIALOGUE ------------										
Parameter/Field	**req**	**req**	**RI**	**ind**	**req**		**req**	**rsp**	**RC**	**cnf**	**req**
Last-Partner-Identifier		C	(=)	(=)							
candidate-initiating-tpsu-titles	U	(=)	(=)	(=)	(=)						
candidate-recipient-tpsu-titles	U	(=)	(=)	(=)	(=)						
Mapping		M									

a) Last-Partner-Identifier: Contains the value of the Correlator of the last TP-BEGIN-DIALOGUE-RI APDU received by the contention-loser. If no dialogue has been established on the association, this parameter/field does not exist.

b) candidate-initiating-tpsu-titles: Lists the possible values of the initiating-tpsu-title on the solicited TP-BEGIN-DIALOGUE request/indication; if the parameter is absent on the request/indication, the initiating-tpsu-title of the solicited dialogue may have any value, or be absent.

c) candidate-recipient-tpsu-titles: Lists the possible values of the recipient-tpsu-title on the solicited TP-BEGIN-DIALOGUE request/indication; if the parameter is absent on the request/indication, the recipient-tpsu-title of the solicited dialogue may have any value, or be absent.

d) Mapping: Indicates the underlying Presentation service to which this AF service is mapped. It has one of the following values:

   1) "dataRI" – Mapped to P-DATA request;

   2) "tokengiveRI" – Mapped to P-TOKEN-GIVE(sync-minor) request.

NOTE – The candidate-initiating/recipient-tpsu-titles parameter names refer to the initiator and recipient of the solicited dialogue establishment. This is opposite to the direction of the solicitation. Thus "candidate-recipient-tpsu-titles" list TPSU-titles that are available at the AEI issuing the AF-SOLICIT-DIALOGUE request.

9.4 Procedures

The following subclauses specify the actions taken upon receipt of the specified AF-Service primitive, TP APDU, or ACSE or CCR Service primitive containing a TP APDU.

NOTE 1 – ACSE and CCR indication and confirm service primitives that do not contain an embedded TP APDU are given to the MACF through the appropriate SACF procedures.

NOTE 2 –TP APDUs not contained in ACSE or CCR Service primitives are contained in Presentation Service primitives.

9.4.1 AF-BEGIN-DIALOGUE request

– Send a TP-BEGIN-DIALOGUE-RI APDU.

9.4.2 TP-BEGIN-DIALOGUE-RI TP APDU

– Issue an AF-BEGIN-DIALOGUE indication.

9.4.3 AF-BEGIN-DIALOGUE response

– Send a TP-BEGIN-DIALOGUE-RC APDU as user data in the service specified by the Mapping parameter.

9.4.4 TP-BEGIN-DIALOGUE-RC TP APDU

– Issue an AF-BEGIN-DIALOGUE confirm with the value of the Mapping parameter set to "dataRI".

9.4.5 AF-BID request

– Send a TP-BID-RI APDU.

9.4.6 TP-BID-RI TP APDU

– Issue an AF-BID indication.

9.4.7 AF-BID response

– Send a TP-BID-RC APDU.

9.4.8 TP-BID-RC TP APDU

– Issue an AF-BID confirm.

9.4.9 AF-END-DIALOGUE request

– Send a TP-END-DIALOGUE-RI APDU.

9.4.10 TP-END-DIALOGUE-RI TP APDU

– Issue an AF-END-DIALOGUE indication.

9.4.11 AF-END-DIALOGUE response

– Send a TP-END-DIALOGUE-RC APDU.

9.4.12 TP-END-DIALOGUE-RC TP APDU

– Issue an AF-END-DIALOGUE confirm.

9.4.13 AF-U-ERROR request

– Send a TP-U-ERROR-RI APDU.

9.4.14 TP-U-ERROR-RI TP APDU

– Issue an AF-U-ERROR indication.

9.4.15 AF-U-ERROR response

– Send a TP-U-ERROR-RC APDU.

9.4.16 TP-U-ERROR-RC TP APDU

– Issue an AF-U-ERROR confirm.

9.4.17 AF-ABORT request

– Send a TP-ABORT-RI TP APDU as user data in the service specified by the Mapping parameter.

9.4.18 TP-ABORT-RI TP APDU

– Issue an AF-ABORT indication with the value of the Mapping parameter set to "dataRI".

9.4.19 AF-GRANT-CONTROL request

– Send a TP-GRANT-CONTROL-RI APDU.

9.4.20 TP-GRANT-CONTROL-RI TP APDU

– Issue an AF-GRANT-CONTROL indication.

9.4.21 AF-REQUEST-CONTROL request

– Send a TP-REQUEST-CONTROL-RI APDU.

9.4.22 TP-REQUEST-CONTROL-RI TP APDU

– Issue an AF-REQUEST-CONTROL indication.

9.4.23 AF-HANDSHAKE request

– Send a TP-HANDSHAKE-RI APDU.

9.4.24 TP-HANDSHAKE-RI TP APDU

– Issue an AF-HANDSHAKE indication.

9.4.25 AF-HANDSHAKE response

– Send a TP-HANDSHAKE-RC APDU.

9.4.26 TP-HANDSHAKE-RC TP APDU

– Issue an AF-HANDSHAKE confirm.

9.4.27 AF-HANDSHAKE-AND-GRANT-CONTROL request

– Send a TP-HANDSHAKE-AND-GRANT-CONTROL-RI APDU.

9.4.28 TP-HANDSHAKE-AND-GRANT-CONTROL-RI TP APDU

– Issue an AF-HANDSHAKE-AND-GRANT-CONTROL indication.

9.4.29 AF-HANDSHAKE-AND-GRANT-CONTROL response

– Send a TP-HANDSHAKE-AND-GRANT-CONTROL-RC APDU.

9.4.30 TP-HANDSHAKE-AND-GRANT-CONTROL-RC TP

– Issue an AF-HANDSHAKE-AND-GRANT-CONTROL confirm.

9.4.31 AF-DEFER request

– Send a TP-DEFER-RI APDU.

9.4.32 TP-DEFER-RI TP APDU

– Issue an AF-DEFER indication.

9.4.33 AF-BEGIN-TRANSACTION request

- Send a TP-BEGIN-TRANSACTION-RI APDU as User Data of a C-BEGIN request.

9.4.34 C-BEGIN indication

The TP-BEGIN-TRANSACTION-RI APDU is received as User Data of the C-BEGIN indication:

- issue an AF-BEGIN-TRANSACTION indication.

9.4.35 AF-PREPARE request

- Send a TP-PREPARE-RI APDU as User Data of a C-PREPARE request.

9.4.36 C-PREPARE indication

The TP-PREPARE-RI APDU is received as User Data of the C-PREPARE indication:

- issue an AF-PREPARE indication.

9.4.37 AF-REPORT request

a) If the value of the Mapping parameter is not "recoverCommitRI" or the Recovery-Context-Handle parameter is not present:

   - send a TP-REPORT-RI APDU as user data in the service specified by the Mapping parameter.

b) If the value of the Mapping parameter is "recoverCommitRI" and the Recovery-Context-Handle parameter is present:

   - send a TP-RECOVER-RI APDU followed by a TP-REPORT-RI APDU as user data of a C-RECOVER (commit) request.

9.4.38 TP-REPORT-RI TP APDU

- Issue an AF-REPORT indication with the Mapping parameter set to "dataRI".

9.4.39 AF-ABORT-AND-REPORT request

a) If the value of the Mapping parameter is "dataRI":

   - send a TP-ABORT-AND-REPORT-RI APDU.

b) Otherwise:

   - send a TP-REPORT-RI APDU followed by a TP-ABORT-RI APDU with the value of the Type field set to "user" as user data in the service specified by the Mapping parameter.

9.4.40 TP-ABORT-AND-REPORT-RI TP APDU

- Issue an AF-ABORT-AND-REPORT indication with the Mapping parameter set to "dataRI".

9.4.41 AF-EARLY-EXIT request

- Send a TP-EARLY-EXIT-RI APDU as user data of a C-ROLLBACK request.

9.4.42 AF-EARLY-EXIT response

- Send a TP-EARLY-EXIT-RC APDU as user data of a C-ROLLBACK response.

9.4.43 AF-RECOVER request

- Send a TP-RECOVER-RI APDU as user data of a C-RECOVER request.

9.4.44 C-RECOVER indication

a) If both a TP-RECOVER-RI APDU and TP-REPORT-RI APDU are received as User Data of the C-RECOVER indication:

   - issue an AF-REPORT indication with the value of the Mapping parameter set to "recoverCommitRI".

b) If only a TP-REPORT-RI APDU is received as User Data of the C-RECOVER indication:

   - issue an AF-REPORT indication with the value of the Mapping parameter set to "recoverCommitRI".

c) If only a TP-RECOVER-RI APDU is received as User Data of the C-RECOVER indication:

   - issue an AF-RECOVER indication.

9.4.45 A-ABORT indication

The TP-ABORT-RI APDU is received as User Information of the A-ABORT indication:

- issue an AF-ABORT indication with the Mapping parameter set to "abortRI".

9.4.46 C-ROLLBACK indication

a) If both a TP-ABORT-RI APDU and a TP-REPORT-RI APDU are received as User Data:
   - issue an AF-ABORT-AND-REPORT indication with the value of the:
     1) Heuristic-Report, Diagnostics, Severity and Completion-Data parameters set to those of the TP-REPORT-RI TP APDU;
     2) Mapping parameter set to "rollbackRI".

b) If only a TP-ABORT-RI APDU is received as User Data:
   - issue an AF-ABORT indication with the value of the Mapping parameter set to "rollbackRI".

c) If only a TP-REPORT-RI APDU is received as User Data:
   - issue an AF-REPORT indication with the value of the Mapping parameter set to "rollbackRI".

d) If a TP-BEGIN-DIALOGUE-RC APDU is received as User Data:
   - issue an AF-BEGIN-DIALOGUE confirm with the value of the Mapping parameter set to "rollbackRI".

e) If a TP-EARLY-EXIT-RI APDU is received as User Data:
   - issue an AF-EARLY-EXIT indication.

9.4.47 C-ROLLBACK confirm

For all the service primitives issued in this procedure, if a TP-REPORT-RI APDU is received as User Data, the Severity, Diagnostic and Completion Data parameters are set to those of the TP-REPORT-RI APDU. If no TP-REPORT-RI APDU is received and these parameters are mandatory on the service primitive, they shall have null values on the service primitive:

a) If both a TP-ABORT-RI APDU and a TP-REPORT-RI APDU are received as User Data:
   - issue an AF-ABORT-AND-REPORT indication with the value of the:
     1) Heuristic-Report, Diagnostics, Severity and Completion-Data parameters set to those of the TP-REPORT-RI TP APDU;
     2) Mapping parameter set to "rollbackRC".

b) If only a TP-ABORT-RI APDU is received as User Data:
   - issue an AF-ABORT indication with the value of the Mapping parameter set to "rollbackRC".

c) If only a TP-REPORT-RI APDU is received as User Data:
   - issue an AF-REPORT indication with the value of the Mapping parameter set to "rollbackRC".

d) If a TP-BEGIN-DIALOGUE-RC APDU is received as User Data:
   - issue an AF-BEGIN-DIALOGUE confirm with the value of the Mapping parameter set to "rollbackRC".

e) If a TP-EARLY-EXIT-RC APDU is received as User Data:
   - issue an AF-EARLY-EXIT confirm.

9.4.48 AF-NOCHANGE request

- Send a TP-NEXT-TID-RI APDU as user data of a C-NOCHANGE request. The Confirmation parameter of the C-NOCHANGE request shall be set to "result-requested".

9.4.49 C-NOCHANGE indication

The TP-NEXT-TID-RI APDU is received as User Data of the C-NOCHANGE indication:

- issue an AF-NOCHANGE indication.

9.4.50 C-NOCHANGE confirm

The TP-ABORT-RI APDU is received as User Data of the C-NOCHANGE confirm:

– issue an AF-ABORT indication with the value of the Mapping parameter set to "nochangeRC".

9.4.51 C-COMMIT indication

The TP-ABORT-RI APDU is received as User Data of the C-COMMIT indication:

– issue an AF-ABORT indication with the value of the Mapping parameter set to "commitRI".

9.4.52 C-COMMIT confirm

a) If both a TP-ABORT-RI APDU and a TP-REPORT-RI APDU are received as User Data:

   – issue an AF-ABORT-AND-REPORT indication with the value of the:

      1) Heuristic-Report and Completion-data parameters set to those of the TP-REPORT-RI TP APDU;

      2) Mapping parameter set to "commitRC".

b) If only a TP-ABORT-RI APDU is received as User Data:

   – issue an AF-ABORT indication with the value of the Mapping parameter set to "commitRC".

c) If only a TP-REPORT-RI APDU is received as User Data:

   – issue an AF-REPORT indication with the value of the Mapping parameter set to "commitRC".

9.4.53 C-RECOVER confirm

The TP-REPORT-RI APDU is received as User Data of the C-RECOVER confirm:

– issue an AF-REPORT indication with the value of the Mapping parameter set to "recoverDoneRC".

9.4.54 P-TOKEN-GIVE (sync-minor) indication

The TP-TOKEN-GIVE-RI APDU is received as User Data of the P-TOKEN-GIVE (sync-minor) indication:

– issue an AF-TOKEN-GIVE indication.

9.4.55 AF-TOKEN-GIVE request

– Send a TP-TOKEN-GIVE-RI APDU as User Data of a P-TOKEN-GIVE (sync-minor) request.

9.4.56 P-TOKEN-PLEASE (sync-minor) indication

a) If a TP-TOKEN-PLEASE-RI APDU is received as User Data of the P-TOKEN-PLEASE (sync-minor) indication:

   – issue an AF-TOKEN-PLEASE indication.

b) If a TP-SOLICIT-DIALOGUE-RI APDU is received as User Data of the P-TOKEN-PLEASE (sync-minor) indication:

   – issue an AF-SOLICIT-DIALOGUE indication.

9.4.57 AF-TOKEN-PLEASE request

– Send a TP-TOKEN-PLEASE-RI APDU as User Data of a P-TOKEN-PLEASE (sync-minor) request.

9.4.58 AF-SOLICIT-DIALOGUE request

– Send a TP-SOLICIT-DIALOGUE-RI TP APDU as user data in the service specified by the Mapping parameter.

9.4.59 TP-SOLICIT-DIALOGUE-RI TP APDU

– Issue an AF-SOLICIT-DIALOGUE indication.

9.4.60 AF-SOLICIT-DIALOGUE response

– Send a TP-SOLICIT-DIALOGUE-RC APDU.

9.4.61 TP-SOLICIT-DIALOGUE-RC TP APDU

- Issue an AF-SOLICIT-DIALOGUE confirm.

9.5 Mapping

All TP APDUs are transferred as specified in clauses 9, 10, and 11, either as non-embedded TP APDUs or embedded in user data of other ASE service primitives.

The TP APDUs are carried by CCR, ACSE, or Presentation services as shown in Table 39, except for the case of concatenation (see 10.7).

Table 39/X.862 – CCR, ACSE and Presentation services carrying TP APDUs

TP APDUs	CCR, ACSE and Presentation Services[a)]
TP-BEGIN-DIALOGUE-RI	P-DATA req
TP-BEGIN-DIALOGUE-RC	P-DATA req, or C-ROLLBACK req, or C-ROLLBACK rsp
TP-BID-RI/RC	P-DATA req
TP-END-DIALOGUE-RI/RC	P-DATA req
TP-ABORT-RI	P-DATA req, or C-ROLLBACK req, or C-ROLLBACK rsp, or C-COMMIT req, or C-COMMIT rsp, or A-ABORT req
TP-U-ERROR-RI/RC	P-DATA req
TP-GRANT-CONTROL-RI	P-DATA req
TP-REQUEST-CONTROL-RI	P-DATA req
TP-HANDSHAKE-RI/RC	P-DATA req
TP-HANDSHAKE-AND-GRANT-CONTROL-RI/RC	P-DATA req
TP-BEGIN-TRANSACTION-RI	C-BEGIN req
TP-PREPARE-RI	C-PREPARE req
TP-NEXT-TID-RI	AF-NOCHANGE req
TP-DEFER-RI	P-DATA req
TP-REPORT-RI	C-ROLLBACK req, or C-ROLLBACK rsp, or C-COMMIT rsp, or C-RECOVER (done) rsp, or P-DATA req, or C-NOCHANGE rsp
TP-TOKEN-GIVE-RI	P-TOKEN-GIVE (sync-minor) req
TP-TOKEN-PLEASE-RI	P-TOKEN-PLEASE (sync-minor) req
TP-RECOVER-RI	C-RECOVER req
TP-INITIALIZE-RI	A-ASSOCIATE req
TP-INITIALIZE-RC	A-ASSOCIATE rsp
TP-SOLICIT-DIALOGUE-RI	P-DATA req, or P-TOKEN-GIVE (sync-minor) req
TP-SOLICIT-DIALOGUE-RC	P-DATA req

[a)] TP APDUs shown as mapping to the P-DATA Service may have a different mapping depending on the combined set of concatenation rules used in the SAO.

10 SACF description

10.1 Introduction

The SACF procedures (described in 10.2 through 10.6) handle the queuing, bidding, and token control related to dialogue/channel establishment as well as the management of the association while not in use by a dialogue or channel. When unqualified, the term SACF refers to these procedures.

Dashed lists are used exclusively to present the actions.

The rules associated with the concatenator part of the SACF are described in 10.7.

The router part of the SACF is described in 10.8.

10.2 SACF states

The SACF has one of the following states at all times. These states are visible only to the SACF. The SACF state is changed either upon receipt of a relevant AF, CCR, or SAF service primitive or, spontaneously, upon certain SACF internal events. When the SAO is newly created, the SACF is in the FREE state. The SACF states are:

a) FREE

This state is used when the SAO is available to be *attached* to a PM for a dialogue or channel. It is therefore in the pool of free associations. When the association is newly established, the SACF is in the FREE state.

b) STRAY

This state is used to filter service primitives that may be received between the receipt of an AF-BEGIN-DIALOGUE request and the receipt of the corresponding AF-BEGIN-DIALOGUE confirm in the case where there is no bidding.

c) BIDDING (contention-loser only)

This state is used to filter service primitives which may be received between the receipt of an AF-BEGIN-DIALOGUE request and the receipt of an AF-BID confirm and the SACF bids. In the case of a dialogue, this state only applies to a subordinate dialogue.

d) BID CONFIRM RECEIVED (contention-loser only)

This state is used to detect protocol errors which become manifest by receiving an indication or confirm primitive between the receipt of an AF-BID confirm and an AF-BEGIN-DIALOGUE confirm. In the case of a dialogue, this state only applies to a subordinate dialogue.

e) BID INDICATION RECEIVED (contention-winner only)

This state is used to detect protocol errors which become manifest by receiving an indication or confirm primitive between the receipt of an AF-BID indication and an AF-BEGIN-DIALOGUE indication. In the case of a dialogue, this state only applies to the superior dialogue.

f) BUSY

This state is used to allow pass-through of service primitives that may be received between the receipt of:

1) an AF-BEGIN-DIALOGUE indication by a contention-loser, or by a contention-winner without having issued an AF-BEGIN-DIALOGUE request; or

2) an AF-BEGIN-DIALOGUE confirm with a valid Correlator; or

3) a C-ROLLBACK indication or confirm after an AF-BEGIN-DIALOGUE request with the Confirmation parameter set to "negative" has been issued and an AF-BEGIN-DIALOGUE confirm has not been received;

and the issuance of an SAF-DETACH-ASSOCIATION request.

g) CLEANUP ROLLBACK INDICATION EXPECTED (superior only)

This state is used when a *rollback indication* is expected outside of a dialogue and must be acknowledged with a *rollback response*.

h) CLEANUP BEGIN INDICATION EXPECTED (subordinate only)

This state is used when a C-BEGIN indication is expected outside of a dialogue and therefore the atomic action branch must be rolled back.

i) CLEANUP ROLLBACK CONFIRM EXPECTED (subordinate only)

This state is used when a *rollback confirm* is expected outside of a dialogue. If, however, a *rollback indication* is received, i.e. a rollback collision occurs, it is the responsibility of the SACF to resend the semantics of the previous *rollback request* in the *rollback response*, using the appropriate AF-service, if necessary.

j) SOLICITING (contention-loser only)

This state is used when a dialogue solicitation has been sent on the association but no reply has been received.

k) SOLICITED (contention-winner only)

This state is used after a dialogue solicitation has been received on the association. In all other respects, the SAO is available as if it were in the FREE state.

10.3 Service definitions for SAF services

10.3.1 SAF-DETACH-ASSOCIATION request

The MACF uses this service to notify the SACF to *detach* from the association.

When the request service primitive is referenced in the procedure clauses, the value listed in the parenthetical argument is the Status parameter.

This is an unconfirmed service.

Table 40 gives the parameter of this primitive:

a) **Status**: Indicates the status of the association that is being *detached*. The values are:

   1) "free" – The association is available for assignment to another dialogue.

   2) "rollback-indication-expected" – The association is in a state where the partner TPPM is expected to issue a *rollback request* and the corresponding *rollback indication* is expected. Once the *rollback indication* is received, the association will be available for assignment to another dialogue.

   3) "rollback-confirm-expected" – A *rollback confirm* is expected. Once the *rollback confirm* is received, the association is available for assignment. If a *rollback indication* is received, the AF-service corresponding to the previous *rollback request* is re-issued as a *rollback response*. If the previous *rollback request* was not an AF-service, a C-ROLLBACK response is issued. An AF-ABORT (user, dataRI) indication may also be received since this may be an alternative to the C-BEGIN indication. When the AF-ABORT (user, dataRI) indication is received, the association is available.

   4) "begin-indication-expected" – A C-BEGIN is expected to be received on this association. Once the C-BEGIN indication is received, a C-ROLLBACK request is issued to roll back the branch. Only when the *rollback confirm* is received is the association available.

   5) "begin-fear" – a C-BEGIN indication may be received in the future due to one of the following conditions:

      i) the Unchained Transactions functional unit is selected and the dialogue was terminated by an AF-ABORT (user, dataRI) request, an AF-END-DIALOGUE request with the Confirmation parameter set to "false", or an AF-BEGIN-DIALOGUE (rejected(user/provider)) response; or

      ii) the *dialogue is chaining*, *rollback reporting has completed*, and an AF-ABORT (user, dataRI) request was issued while in the DECIDED (rollback) state.

Table 40/X.862 – SAF-DETACH-ASSOCIATION parameter

SAF-DETACH-ASSOCIATION	
Parameter	**req**
Status	M

10.3.2 SAF-ASSOCIATION-LOST indication

The SACF uses this service to notify the MACF that it is *detached* from the association.

10.3.3 SAF-SOLICIT-DIALOGUE request/indication/response/confirm

This is used by the contention-loser to solicit the establishment of a dialogue by the contention-winner.

This is an optionally confirmed service. It is confirmed only when the solicitation is explicitly rejected.

Table 41 gives the parameters of this service:

a) candidate-initiating-tpsu-titles: This parameter lists the possible values of the initiating-tpsu-title on the solicited TP-BEGIN-DIALOGUE request/indication; if the parameter is absent on the request/indication, the initiating-tpsu-title of the solicited dialogue may have any value, or be absent.

b) candidate-recipient-tpsu-titles: This parameter lists the possible values of the recipient-tpsu-title on the solicited TP-BEGIN-DIALOGUE request/indication; if the parameter is absent on the request/indication, the recipient-tpsu-title of the solicited dialogue may have any value, or be absent.

NOTE – The parameter names refer to the initiation and recipient of the solicited dialogue establishment. This is opposite to the direction of the solicitation.

Table 41/X.862 – SAF-SOLICIT-DIALOGUE parameter

SAF-SOLICIT-DIALOGUE				
Parameter	**req**	**ind**	**rsp**	**cnf**
candidate-initiating-tpsu-titles	U	C(=)		
candidate-recipient-tpsu-titles	U	C(=)		

10.4 Procedures for SAF primitives

10.4.1 SAF-DETACH-ASSOCIATION request

If the value of the status parameter is:

a) "free" or "begin-fear"; and:

   1) a queue exists:

      – continue;

   2) no queue exists:

      – enter the FREE state;

      – issue an AF-TOKEN-GIVE (regular) request if the SACF is a contention-loser and the *token* is owned;

b) "rollback-indication-expected":

   – enter the CLEANUP ROLLBACK INDICATION EXPECTED state.

c) "rollback-confirm-expected":

   – enter the CLEANUP ROLLBACK CONFIRM EXPECTED state.

d) "begin-indication-expected":

   – enter the CLEANUP BEGIN INDICATION EXPECTED state.

Always:

– *detach* the MACF from the association.

10.4.2 SAF-SOLICIT-DIALOGUE request

The SACF shall be in the FREE state and shall be the contention-loser.

An AF-SOLICIT-DIALOGUE request shall be issued.

Always:

– enter the SOLICITING state.

If the token is owned:

– issue an AF-SOLICIT-DIALOGUE (tokengiveRI) request.

If the token is not owned:

– issue an AF-SOLICIT-DIALOGUE (dataRI) request.

10.4.3 SAF-SOLICIT-DIALOGUE response

The SACF shall be in the SOLICITED state and shall be the contention-winner:

– issue an AF-SOLICIT-DIALOGUE response;

– enter the FREE state.

10.5 Procedures for TP-ASE, CCR, ACSE, and Presentation Service primitives

Service primitives received from the MACF are assumed valid, i.e. issued in accordance with the procedures specified in clause 11.

10.5.1 AF-BEGIN-DIALOGUE request

This service primitive is received while the SACF is in the FREE or SOLICITED state, and the association is compatible with the dialogue or channel.

If the SACF is a contention-loser and one or more of the following is true:

a) the bid-mandatory field of the TP-INITIALIZE-RI APDU was "true" for this association;

b) the last time this SAO was detached from the MACF, this was done by an SAF-DETACH-ASSOCIATION request which had the Type parameter set to "begin-fear"; or

c) according to a local decision, the SACF will bid,

then:

a) Formulate the parameters of the AF-BID request with the value of the:

   1) Last-Partner-Identifier parameter:

      i) omitted if no AF-BEGIN-DIALOGUE indication has been received on this association since it was established; or

      ii) set to the Correlator of the most recently received AF-BEGIN-DIALOGUE indication, otherwise.

   2) CCR-Token-Requested parameter set to either:

      i) "true" if the Commit or Recovery Functional Unit is selected; or

      ii) "false" otherwise;

b) Take the following actions:

   – issue an AF-BID request;

   – *establish a queue*;

   – enter the BIDDING state.

If no AF-BID request was issued:

– enter the STRAY state.

If the SACF is a contention-winner and the last time this SAO was detached from the MACF, this was done by an SAF-DETACH-ASSOCIATION request which had the Type parameter set to "begin-fear":

– *establish a queue*.

Always take the following actions:

a) Formulate the parameters of the AF-BEGIN-DIALOGUE request with the value of the:

   1) Last-Partner-Identifier parameter:

      i) Omitted if any of the following are true:

         - the SACF is a contention-winner;

         - an AF-BID request was issued; or

         - no AF-BEGIN-DIALOGUE indication has been received on this association since it was established.

      ii) Set to the Correlator of the most recently received AF-BEGIN-DIALOGUE indication, otherwise.

   2) Correlator parameter set to a value which is unique within the scope of the association.

b) If a queue exists:

   – queue the AF-BEGIN-DIALOGUE request.

c) If a queue does not exist:

   – pass through the AF-BEGIN-DIALOGUE request.

10.5.2 AF-BEGIN-DIALOGUE indication

If the SACF is the contention-winner and the Bid-Mandatory field of the TP-INITIALIZE-RI APDU was set to "false", and either:

a) the SACF is in the FREE state and the *last partner identifier is not valid*; or

b) the SACF is in the STRAY state:

   – continue.

If the SACF is the contention-loser and in the STRAY or BIDDING states:

– issue an SAF-ASSOCIATION-LOST indication if no SAF-DETACH-ASSOCIATION request was received;

– *discard the queue* (if any).

NOTE 1 – It is a local matter whether the TPPM retries establishing a dialogue or channel on another association; however, in the case where the SAF-ASSOCIATION-LOST indication is not issued because an SAF-DETACH-ASSOCIATION request was already received, retrying is recommended.

If all of the following conditions are true:

a) the SACF is the contention-winner in the FREE state and the *last partner identifier is valid*;

b) the Commit or Recovery functional unit is selected;

c) the Bid-Mandatory field of the TP-INITIALIZE-RI APDU was set to "false";

d) the association has not been reserved for other use; and

e) the *token* is owned by the SACF,

then:

– issue an AF-TOKEN-GIVE (keep) request.

NOTE 2 – If the Commit functional unit is not supported, this service primitive need not be generated, as the dialogue will be rejected by the MACF.

If the SACF is one of the following:

a) the contention-winner and either in the FREE state, the *last partner identifier is valid*, the Bid-Mandatory field of the TP-INITIALIZE-RI APDU was set to "false", and the association has not been reserved for other use, or in the BID INDICATION RECEIVED state and an AF-BID response has been sent; or

b) the contention-loser and in the FREE, SOLICITING, STRAY, or BIDDING state,

then:

– create a new MACF, if this is a dialogue;

– *attach* to the CPM, if this is a channel;

– enter the BUSY state;

– pass the service primitive through.

If the SACF is the contention-winner and in the FREE state and the Bid-Mandatory field of the TP-INITIALIZE-RI APDU was set to "false", the *last partner identifier is valid*, and the association has been reserved for other use:

– issue an AF-BEGIN-DIALOGUE (rejected(provider), dataRI) response with the Diagnostic parameter set to "association-reserved".

10.5.3 AF-BEGIN-DIALOGUE response

The Correlator parameter of the AF-BEGIN-DIALOGUE response is set to the value of the parameter on the previous AF-BEGIN-DIALOGUE indication:

– *discard any PDUs in the separator*, if the SACF is in the BUSY state and the Mapping parameter is set to "rollbackRI";

– pass the service primitive through.

10.5.4 AF-BEGIN-DIALOGUE confirm

If the SACF is in the FREE, SOLICITED or BIDDING state and the Mapping parameter is set to "dataRI":

– continue.

If the SACF is in the STRAY state and the Mapping parameter is set to "dataRI" and the Correlator does not match the Correlator of the previous AF-BEGIN-DIALOGUE request:

– continue.

If the SACF is in the STRAY or BID CONFIRM RECEIVED state and the Correlator matches the Correlator of the previous AF-BEGIN-DIALOGUE request and the Diagnostic parameter is either absent or set to a value different from "association-reserved":

– enter the BUSY state;

– pass the service primitive through.

If the SACF is the contention-loser in the STRAY state and the Correlator matches the Correlator of the previous AF-BEGIN-DIALOGUE request and the Diagnostic parameter is set to "association-reserved" and the Mapping parameter is set to "dataRI":

– enter the BUSY state;

– pass the service primitive through.

10.5.5 AF-BID indication

The SACF shall be the contention-winner.

If either of the following is true:

a) the SACF is in the FREE state and the *last partner identifier is not valid*; or

b) the SACF is in the STRAY or CLEANUP ROLLBACK INDICATION EXPECTED state,

then:

– continue.

If the SACF is in the FREE state and the *last partner identifier is valid* and all of the following conditions are met:

a) the value of the CCR-Token-Requested parameter is set to "true"; and

b) the SACF does not have the *token*; and

NOTE 1 – The situation that the value of the CCR-Token-Requested parameter is set to "true" and the TPPM does not have the *token* may arise if a P-TOKEN-GIVE (sync-minor) request crosses an unconfirmed dialogue ending APDU, and the AF-BID request was issued prior to receiving the P-TOKEN-GIVE (sync-minor) indication;

c) according to a local decision, the SACF intends to issue an AF-BID (accepted) response only after the *token* arrives,

then:

– enter the BID INDICATION RECEIVED state.

If the SACF is in the FREE state and none of the preceding conditions apply:

a) Always:

   – issue an AF-BID response with the Result parameter set based on a local decision.

   NOTE 2 – The decision on whether the Result parameter of an AF-BID response is set to "rejected" or to "accepted" will be made depending on whether the association on which the AF-BID indication was received has been reserved for other purposes or not.

b) If the Result parameter is set to "rejected":

   – continue.

c) If the Result parameter is set to "accepted":

   – enter the BID INDICATION RECEIVED state;

   – issue an AF-TOKEN-GIVE (regular) request if the CCR-Token-Requested parameter on the AF-BID indication is set to "true", and the SACF has the *token*.

10.5.6 AF-BID confirm

If the SACF is in the BIDDING state and the result parameter is "accepted":

– *Flush the queue* up to and excluding a C-BEGIN request.

– Enter the FREE state if an SAF-DETACH-ASSOCIATION request was received.

  NOTE 1 – The MACF procedures in clause 11 are designed in such a way that an SAF-DETACH-ASSOCIATION request will never be received after a C-BEGIN request was queued and before the queue is entirely flushed.

– Enter the BID CONFIRM RECEIVED state if no SAF-DETACH-ASSOCIATION request was received.

If the SACF is in the BIDDING state and the Result parameter is "rejected":

– enter the FREE state;

– issue an SAF-ASSOCIATION-LOST indication if no SAF-DETACH-ASSOCIATION request was received;

– *discard the queue*.

NOTE 2 – It is a local matter whether the TPPM retries establishing a dialogue or channel on another association; however, in the case where the SAF-ASSOCIATION-LOST indication is not issued because an SAF-DETACH-ASSOCIATION request was already received, retrying is recommended.

10.5.7 AF-END-DIALOGUE request

If a queue exists:

- queue the service primitive.

If no queue exists:

- pass the service primitive through.

10.5.8 AF-END-DIALOGUE indication

If the SACF is in the FREE, BIDDING, SOLICITING or STRAY state:

- continue.

If the SACF is in the BUSY state:

- pass the service primitive through.

10.5.9 AF-END-DIALOGUE confirm

If the SACF is in the FREE, BIDDING, SOLICITING or STRAY state:

- continue.

If the SACF is in the BUSY state:

- pass the service primitive through.

10.5.10 AF-U-ERROR request

If a queue exists:

- queue the service primitive.

If no queue exists:

- pass the service primitive through.

10.5.11 AF-U-ERROR indication

If the SACF is in the FREE, BIDDING, SOLICITING, STRAY, or CLEANUP ROLLBACK INDICATION EXPECTED state:

- continue.

If the SACF is in the BUSY state:

- pass the service primitive through.

10.5.12 AF-U-ERROR confirm

If the SACF is in the FREE, BIDDING, SOLICITING or STRAY state:

- continue.

If the SACF is in the BUSY state:

- pass the service primitive through.

10.5.13 AF-ABORT request

If a queue exists and the value of the Mapping parameter is not "abortRI":

- queue the service primitive.

If a queue does not exist or the value of the Mapping parameter is "abortRI":

- *discard any PDUs in the separator*, if the Mapping parameter is "abortRI" or "rollbackRI";
- pass the service primitive through.

NOTE – If the Mapping parameter is "abortRI", the association will cease to exist when the A-ABORT request is issued.

10.5.14 AF-ABORT (provider, abortRI) indication

NOTE – Upon receipt of an AF-ABORT (provider, abortRI) indication, the association ceases to exist.

If the SACF is in the FREE, SOLICITING, SOLICITED, BID INDICATION RECEIVED, CLEANUP ROLLBACK INDICATION EXPECTED, or CLEANUP BEGIN INDICATION RECEIVED state:

– continue.

If the SACF is in the STRAY, BIDDING, BID CONFIRM RECEIVED, or BUSY state:

– pass the service primitive through if no SAF-DETACH-ASSOCIATION request was received;

– *discard the queue*.

10.5.15 AF-ABORT (user, dataRI) indication

If the SACF is in the FREE, BIDDING, SOLICITING, STRAY, or CLEANUP ROLLBACK CONFIRM EXPECTED state:

– continue.

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.16 A-ABORT request

If the SACF is in the STRAY, BIDDING, SOLICITING, SOLICITED, BID CONFIRM RECEIVED, BUSY, CLEANUP ROLLBACK INDICATION EXPECTED, or CLEANUP BEGIN INDICATION EXPECTED state:

– *discard any PDUs in the separator*;

– *discard the queue*;

– pass the service primitive through.

NOTE – Upon issuance of this service primitive, the association ceases to exist.

10.5.17 A-RELEASE (Result = affirmative) response

If the SACF is in the STRAY, BIDDING, SOLICITING, BID CONFIRM RECEIVED, or BUSY state:

– *discard the queue*;

– pass the service primitive through.

NOTE – Upon issuance of this service primitive, the association ceases to exist.

10.5.18 A-[P-]ABORT indication or A-RELEASE (Result = affirmative) confirm

If the SACF is in the BIDDING, STRAY, SOLICITING, BID CONFIRM RECEIVED, or BUSY state:

– pass the service primitive through, if no SAF-DETACH-ASSOCIATION request was received.

10.5.19 AF-GRANT-CONTROL request

If a queue exists:

– queue the service primitive.

If no queue exists:

– pass the service primitive through.

10.5.20 AF-GRANT-CONTROL indication

If the SACF is in the FREE, SOLICITING, BIDDING, or STRAY state:

– continue.

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.21 AF-REQUEST-CONTROL request

If a queue exists:

- queue the service primitive.

If no queue exists:

- pass the service primitive through.

10.5.22 AF-REQUEST-CONTROL indication

If the SACF is in the FREE, SOLICITING, BIDDING, or STRAY state:

- continue.

If the SACF is in the BUSY state:

- pass the service primitive through.

10.5.23 AF-HANDSHAKE request

If a queue exists:

- queue the service primitive.

If no queue exists:

- pass the service primitive through.

10.5.24 AF-HANDSHAKE indication

If the SACF is in the FREE, SOLICITING, BIDDING, or STRAY state:

- continue.

If the SACF is in the BUSY state:

- pass the service primitive through.

10.5.25 AF-HANDSHAKE confirm

If the SACF is in the FREE, SOLICITING, BIDDING, or STRAY state:

- continue.

If the SACF is in the BUSY state:

- pass the service primitive through.

10.5.26 AF-HANDSHAKE-AND-GRANT-CONTROL request

If a queue exists:

- queue the service primitive.

If no queue exists:

- pass the service primitive through.

10.5.27 AF-HANDSHAKE-AND-GRANT-CONTROL indication

If the SACF is in the FREE, BIDDING, SOLICITING or STRAY state:

- continue.

If the SACF is in the BUSY state:

- pass the service primitive through.

10.5.28 AF-HANDSHAKE-AND-GRANT-CONTROL confirm

If the SACF is in the FREE, BIDDING, SOLICITING or STRAY state:

- continue.

If the SACF is in the BUSY state:

- pass the service primitive through.

10.5.29 AF-DEFER request

If a queue exists:

– queue the service primitive.

If no queue exists:

– pass the service primitive through.

10.5.30 AF-DEFER indication

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.31 AF-PREPARE request

If a queue exists:

– queue the service primitive.

If no queue exists:

– pass the service primitive through.

10.5.32 AF-PREPARE indication

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.33 AF-REPORT (commitRC) indication, or AF-REPORT (recoverDoneRC) indication

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.34 C-BEGIN request or AF-BEGIN-TRANSACTION request

If the SACF does not have the *token* and is in the:

a) STRAY or BID CONFIRM RECEIVED state:

   – *establish a queue*;

   – queue the service primitive.

b) BIDDING state:

   – queue the service primitive.

NOTE – Only under these conditions will the SACF queue for the *token* on a dialogue. Under other conditions, the TPSUI is assumed to have the *token* (see Annex B). For conditions where the SACF will queue for the *token* on a channel, see 10.5.53.

If the SACF has the *token*:

– pass the service primitive through.

10.5.35 C-BEGIN indication or AF-BEGIN-TRANSACTION indication

If the SACF is in the FREE or SOLICITING state and the last time this SAO was detached from the MACF this was done by an SAF-DETACH-ASSOCIATION request which had the Type parameter set to "begin-fear", or if the SACF is in the CLEANUP BEGIN INDICATION EXPECTED state:

– *discard any PDUs in the separator*;

– issue a C-ROLLBACK request;

– enter the CLEANUP ROLLBACK CONFIRM EXPECTED state.

If the SACF is in the BUSY state:

– pass the service primitive through.

If the SACF is in the BIDDING or STRAY state and the last time this SAO was detached from the MACF, this was done by an SAF-DETACH-ASSOCIATION request which had the Type parameter set to "begin-fear":

– Issue an SAF-ASSOCIATION-LOST indication if no SAF-DETACH-ASSOCIATION request was received.

– *Discard the queue*.

NOTE – It is a local matter whether the TPPM retries establishing a dialogue or channel on another association; however, in the case where the SAF-ASSOCIATION-LOST indication is not issued because an SAF-DETACH-ASSOCIATION request was already received, retrying is recommended.

– *Discard any PDUs in the separator*.

– Issue a C-ROLLBACK request.

– Enter the CLEANUP ROLLBACK CONFIRM EXPECTED state.

10.5.36 C-BEGIN confirm

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.37 C-READY indication

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.38 C-COMMIT indication or C-COMMIT+C-BEGIN indication

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.39 AF-ABORT (user, commitRI) indication or AF-ABORT (user, commitRC) indication

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.40 C-COMMIT confirm

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.41 AF-ABORT-AND-REPORT (commitRC) indication

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.42 C-ROLLBACK request

If a queue exists:

– queue the service primitive.

If no queue exists:

– *discard any PDUs in the separator*;

– pass the service primitive through.

10.5.43 C-ROLLBACK indication

NOTE 1 – After receipt of a *rollback response* or a *rollback confirm*, the *token* is set according to the TP/CCR rules described in 8.4.2 and sent to the superior of the branch. Since the SACF is not aware of the superior of the branch, it returns the *token* if the SACF is the contention-loser.

If the SACF is in the BUSY state:

– pass the service primitive through.

If the SACF is in the STRAY or BID CONFIRM RECEIVED state:

- enter the BUSY state;
- pass the service primitive through.

If the SACF is in the:

a) CLEANUP ROLLBACK CONFIRM EXPECTED state:

   1) If the previous request or response was an AF-ABORT (provider, rollbackRI) request with the Diagnostic parameter set to "begin-transaction-reject":

      - issue an AF-ABORT (provider, rollbackRC) request with the Diagnostic parameter set to "begin-transaction-reject".

   2) If the previous request or response was an AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) response:

      - issue an AF-BEGIN-DIALOGUE (rejected(user), rollbackRC) response, with the same values of the corresponding parameters as in the previous AF-BEGIN-DIALOGUE response, except for the Mapping parameter.

   3) If the previous request or response was neither an AF-ABORT (provider, rollbackRI) request with Diagnostic parameter set to "begin-transaction-request", nor an AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) response:

      - issue a C-ROLLBACK response.

      NOTE 2 – The semantic of "abort" or "rejected dialogue" carried on the previous C-ROLLBACK request must be repeated on the C-ROLLBACK response [in a) 2) and 3) above] because the C-ROLLBACK-RI has been suppressed by the Session Layer when the C-ROLLBACK-RI collision occurs.

   4) If the SACF is the contention-loser and owns the *token*:

      - issue an AF-TOKEN-GIVE (regular) request.

   5) Always:

      - enter the FREE state.

b) CLEANUP ROLLBACK INDICATION EXPECTED state:

   1) Always:

      - issue a C-ROLLBACK response;
      - enter the FREE state.

   2) If the SACF is the contention-loser and owns the *token*:

      - issue an AF-TOKEN-GIVE (regular) request.

10.5.44 AF-ABORT-AND-REPORT request or AF-REPORT request

If the SACF is in the BUSY state:

- *discard any PDUs in the separator*, if the Mapping parameter is "rollbackRI";
- pass the service primitive through.

10.5.45 AF-ABORT-AND-REPORT (dataRI) indication or AF-REPORT (user, dataRI) indication

If the SACF is in the BUSY state:

- pass the service primitive through.

10.5.46 AF-ABORT (user/provider, rollbackRI) indication, AF-ABORT-AND-REPORT (rollbackRI) indication, AF-REPORT (rollbackRI) indication or AF-EARLY-EXIT indication

NOTE – After receipt of a *rollback response* or a *rollback confirm*, the *token* is set according to the TP/CCR rules described in 8.4.2 and sent to the superior of the branch. Since the SACF is not aware of the superior of the branch, it returns the *token* if the SACF is the contention-loser.

If the SACF is in the BUSY state:

- pass the service primitive through.

If the SACF is in the STRAY or BID CONFIRM RECEIVED state:

– enter the BUSY state;

– pass the service primitive through.

If the SACF is in the CLEANUP ROLLBACK CONFIRM EXPECTED state and this is an AF-ABORT (user/provider, rollbackRI) indication:

a) Always:

  – issue a C-ROLLBACK response;

  – enter the FREE state.

b) If the SACF is the contention-loser and owns the *token*:

  – issue an AF-TOKEN-GIVE (regular) request.

10.5.47 C-ROLLBACK confirm, AF-REPORT (rollbackRC) indication, AF-ABORT (user/provider, rollbackRC) indication or AF-ABORT-AND-REPORT (rollbackRC) indication

NOTE – After receipt of a *rollback response* or a *rollback confirm*, the *token* is set according to the TP/CCR rules described in 8.4.2 and sent to the superior of the branch. Since the SACF is not aware of the superior of the branch, it returns the *token* if the SACF is the contention-loser.

If the SACF is in the BUSY state:

– pass the service primitive through.

If the SACF is in the STRAY or BID CONFIRM RECEIVED state:

– enter the BUSY state;

– pass the service primitive through.

If the SACF is in CLEANUP ROLLBACK CONFIRM EXPECTED state:

– enter the FREE state;

– issue an AF-TOKEN-GIVE (regular) request, if the SACF is a contention-loser and owns the *token*.

10.5.48 AF-NOCHANGE request or C-NOCHANGE request

If a queue exists:

– queue the service primitive.

If no queue exists:

– pass the service primitive through.

10.5.49 C-NOCHANGE indication or AF-NOCHANGE indication

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.50 C-NOCHANGE confirm

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.51 AF-EARLY-EXIT confirm

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.52 AF-RECOVER indication

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.53 C-RECOVER request or AF-RECOVER request

If the SACF is in BUSY state and:

a) At least one of the two following conditions is true:

   1) the *token* is owned;

   2) a C-RECOVER (ready) indication or an AF-RECOVER (ready) indication was the most recently received primitive (including requests and responses from the MACF) on this association, and this is a C-RECOVER (commit) request:

      – pass the service primitive through.

b) Otherwise:

   – *establish a queue*, if one does not exist;

   – queue the service primitive.

10.5.54 C-RECOVER indication

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.55 C-RECOVER confirm

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.56 U-ASE request

If a queue exists:

– queue the service primitive.

If no queue exists:

– pass the service primitive through.

10.5.57 U-ASE indication

If the SACF is in the FREE, BIDDING, SOLICITING or STRAY state:

– suppress the service primitive.

If the SACF is in the BUSY state:

– pass the service primitive through.

10.5.58 AF-TOKEN-GIVE (regular) indication

If the SACF is the contention-loser in the:

a) FREE, STRAY, SOLICITING or BIDDING state:

   – issue an AF-TOKEN-GIVE (regular) request.

b) BID CONFIRM RECEIVED state:

   – *flush the queue*.

If the SACF is a contention-winner in the:

a) STRAY state, and:

   1) there is a queue:

      – *flush the queue*;

      – enter the FREE state, if an SAF-DETACH-ASSOCIATION request has been received;

   2) there is no queue:

      – continue;

      – enter the FREE state, if an SAF-DETACH-ASSOCIATION request has been received.

b) BID INDICATION RECEIVED state and received an AF-BID indication with the CCR-Token-requested parameter set to "false":

   – continue.

c) FREE or SOLICITED state:

   – continue.

d) BID INDICATION RECEIVED state and received an AF-BID indication with the CCR-Token-requested parameter set to "true", and:

   1) has not issued an AF-BID (accepted) response:

      – issue an AF-BID (accepted) response;

      – issue an AF-TOKEN-GIVE (regular) request;

   2) has issued an AF-BID (accepted) response but has not issued an AF-TOKEN-GIVE (regular) request:

      – issue an AF-TOKEN-GIVE (regular) request.

e) BUSY state and all of the following are true:

   1) the Commit functional unit is selected;

   2) an AF-BEGIN-DIALOGUE indication was received and was not preceded by an AF-BID indication; and

   3) no AF-TOKEN-GIVE (keep) request was issued; then:

      – issue an AF-TOKEN-GIVE (keep) request.

10.5.59 AF-TOKEN-GIVE (keep) indication

If the SACF is a contention-loser and is in the STRAY or BUSY state and the value of the Correlator parameter is equal to the Correlator of the last TP-BEGIN-DIALOGUE-RI APDU sent:

– *flush the queue*, if a queue exists;

– continue.

If the SACF is a contention-loser and is in the CLEANUP ROLLBACK INDICATION EXPECTED or CLEANUP ROLLBACK CONFIRM EXPECTED state and the value of the Correlator parameter is equal to the Correlator of the last TP-BEGIN-DIALOGUE-RI APDU sent:

– *flush the queue*.

If the SACF is a contention-loser and is either:

a) in the FREE or SOLICITING state; or

b) in the STRAY or BIDDING state and the value of the Correlator parameter is not equal to the Correlator of the last TP-BEGIN-DIALOGUE-RI APDU sent,

then:

– issue an AF-TOKEN-GIVE (regular) request.

10.5.60 AF-TOKEN-GIVE (two-way-recovery) request

If the SACF is in BUSY state and:

a) The *token* is owned:

   – pass the service primitive through.

b) The *token* is not owned:

   – *establish a queue*, if one does not exist;

   – queue the service primitive.

10.5.61 AF-TOKEN-GIVE (two-way-recovery) indication

If the association is assigned to a channel and the SACF is in the BUSY state and there is no queue:

– pass the service primitive through.

If the association is assigned to a channel and the SACF is in the BUSY state and there is a queue:

– *flush the queue*.

10.5.62 P-TOKEN-GIVE (sync-minor) indication

If the SACF is the contention-loser in the:

a) FREE, STRAY, SOLICITING or BIDDING state:
   - issue an AF-TOKEN-GIVE (regular) request.

b) BUSY state, the Dialogue functional unit is selected and the U-ASE uses the P-TOKEN-GIVE service:
   - pass the *token* to the U-ASE.

c) BUSY state, the Dialogue functional unit is selected and the U-ASE does not use the P-TOKEN-GIVE service:
   - continue.

If the SACF is the contention-winner in the:

a) STRAY state, and:
   1) there is a queue:
      - *flush the queue*;
      - enter the FREE state if an SAF-DETACH-ASSOCIATION request has been received;
   2) there is no queue:
      - continue.

b) FREE state:
   - continue.

c) BUSY state, the Dialogue functional unit is selected and the U-ASE uses the P-TOKEN-GIVE service:
   - pass the *token* to the U-ASE.

d) BUSY state, the Dialogue functional unit is selected and the U-ASE does not use the P-TOKEN-GIVE service:
   - continue.

10.5.63 AF-TOKEN-PLEASE request

If the SACF is in BUSY state and:

a) A queue exists:
   - queue the service primitive.

b) No queue exists:
   - pass the service primitive through.

10.5.64 AF-TOKEN-PLEASE indication

If the association is assigned to a channel and the SACF is in the:

a) BUSY state:
   - pass the service primitive through.

b) FREE, BIDDING, or STRAY state:
   - continue.

10.5.65 P-TOKEN-PLEASE indication

If the association is assigned to a dialogue and the SACF is in the BUSY state:

- pass the service primitive to the U-ASE,

otherwise:

- continue.

10.5.66 AF-SOLICIT-DIALOGUE indication

The SACF shall be the contention-winner.

If either of the following is true:

a) the SACF is in the FREE state and the *last partner identifier is not valid*; or

b) the SACF is in the STRAY or CLEANUP ROLLBACK INDICATION EXPECTED state,

then:

– continue.

If the SACF is in the FREE state and the *last partner identifier is valid*:

– issue an SAF-SOLICIT-DIALOGUE indication;

– enter the SOLICITED state.

10.5.67 AF-SOLICIT-DIALOGUE confirm

The SACF shall be the contention-loser and shall be in the SOLICITING state.

Always:

– issue an SAF-SOLICIT-DIALOGUE confirm;

– enter the FREE state.

10.5.68 Protocol error

If the SACF is not *attached* to an MACF:

– *discard any PDUs in the separator*;

– *discard the queue*;

– issue an AF-ABORT (provider, abortRI) request, with the Diagnostic parameter set to "protocol-error".

10.5.69 Other service primitives

If the SACF is in the BUSY state, the following service primitives are always passed through:

– AF-END-DIALOGUE response;

– AF-U-ERROR response;

– AF-HANDSHAKE response;

– AF-HANDSHAKE-AND-GRANT-CONTROL response;

– AF-TOKEN-GIVE (regular) request;

– AF-TOKEN-GIVE (keep) request;

– C-BEGIN response;

– C-READY request;

– C-COMMIT request;

– C-COMMIT response;

– C-COMMIT+C-BEGIN request;

– C-ROLLBACK response;

– C-NOCHANGE response;

– AF-EARLY-EXIT request;

– AF-EARLY-EXIT response;

– C-RECOVER response;

– P-TOKEN-GIVE (sync-minor) request;

– A-ABORT request.

10.6 SACF internal events

10.6.1 Unsolicited BID reject

This procedure is entered by a contention-winner SACF that is in the BID INDICATION RECEIVED state and has not yet issued an AF-BID response.

NOTE – The SACF, originally intending to accept the BID, but not having the *token*, delayed responding until the *token* was received. In the meantime, some internal condition has arisen that warrants rejecting the BID:

– issue an AF-BID response with the value of the Result parameter set to "rejected";

– enter the FREE state

10.7 Concatenation

10.7.1 Mapping precedence

This Recommendation defines the rules governing the concatenation of TP APDUs and their mapping onto other services. These concatenation rules do not affect lower layer concatenation mechanisms (e.g. the Session Layer concatenation). Alternative mappings to Presentation Services are sometimes used depending on the APDUs being concatenated.

The mapping of any concatenation sequence involving one or more CCR APDUs shall be as specified in ITU-T Rec. X.852 | ISO/IEC 9805-1. Concatenation sequences involving only TP APDUs shall be mapped to the User data parameter of the P-DATA service with the exception of those TP APDUs for which a mapping to a different Presentation or ACE service is specifically defined in the procedures in clause 9. Concatenations involving U-ASE APDUs and not involving any CCR APDUs shall be mapped to the User data parameter of the P-DATA service unless otherwise specified in the U-ASE specification. Any such other mapping shall not interfere with the operation or semantics of either CCR or TP.

Subclause 9.5 shows the TP mapping of TP APDUs onto underlying services, if the TP concatenation mechanism is not used (basic unconcatenated TP APDU mapping).

10.7.2 Concatenation rules

This subclause specifies the rules for determining valid concatenations of TP, U-ASE, CCR, and ACSE APDUs.

NOTE 1 – This subclause does not specify the complete set of rules for determining valid sequences of TP, U-ASE, CCR, and ACSE APDUs. The ACSE and CCR standards themselves impose constraints which are not repeated here. Also the procedures in clauses 9, 10, and 11 further restrict the valid sequences of TP, U-ASE, CCR, and ACSE APDUs.

a) The APDUs in Table 42 shall not be concatenated with any APDU:

Table 42/X.862 – APDUs that cannot be concatenated

TP-BID-RI
TP-BID-RC
TP-BEGIN-DIALOGUE-RC (rejected)
TP-PREPARE-RI
TP-NEXT-TID-RI
TP-RECOVER-RI
TP-SOLICIT-DIALOGUE-RI
TP-SOLICIT-DIALOGUE-RC
C-ROLLBACK-RI
C-ROLLBACK-RC
C-CANCEL-RI
C-RECOVER-RI
C-RECOVER-RC

b) The APDUs in Table 43 shall be first in a concatenation sequence. These APDUs must begin a concatenation sequence because no prior APDU is possible;

Table 43/X.862 – APDUs that begin a concatenation sequence

TP-BEGIN-DIALOGUE-RI
TP-BEGIN-DIALOGUE-RC (accepted)
C-NOCHANGE-RC

c) The APDUs in Table 44 shall be last in a concatenation sequence. These APDUs must end a concatenated sequence because they are part of a confirmed exchange or because no subsequent APDU is possible.

Table 44/X.862 – APDUs that end a concatenation sequence

TP-END-DIALOGUE-RI
TP-END-DIALOGUE-RC
TP-ABORT-RI
TP-ABORT-AND-REPORT-RI
TP-GRANT-CONTROL-RI
TP-REQUEST-CONTROL-RI
TP-HANDSHAKE-RI
TP-HANDSHAKE-AND-GRANT-CONTROL-RI
C-PREPARE-RI
C-READY-RI
C-NOCHANGE-RI

Based on a local decision, each APDU in Table 44 may be concatenated to an existing concatenation sequence.

d) If the preceding TP-HANDSHAKE-RI or TP-HANDSHAKE-AND-GRANT-CONTROL-RI had the confirmation-urgency field equal to "urgent", or the confirmation-urgency field was absent from the TP-HANDSHAKE-RI, then the corresponding APDU in Table 45 shall be last in a concatenation sequence.

If the preceding TP-HANDSHAKE-RI or TP-HANDSHAKE-AND-GRANT-CONTROL-RI had the confirmation-urgency field equal to "normal", then the corresponding APDU in Table 45 need not end a concatenation sequence.

Based on a local decision, each APDU in Table 45 may be concatenated to an existing concatenation sequence.

Table 45/X.862 – APDUs that conditionally end a concatenation sequence

TP-HANDSHAKE-RC
TP-HANDSHAKE-AND-GRANT-CONTROL-RC

e) The APDUs in Table 46 may be concatenated and need not begin nor end a concatenation sequence. Based on a local decision, each APDU in Table 46 may be concatenated to an existing concatenation sequence.

Table 46/X.862 – APDUs that may be concatenated

TP-U-ERROR-RI
TP-U-ERROR-RC
TP-TOKEN-GIVE-RI
TP-TOKEN-PLEASE-RI
TP-DEFER-RI
TP-ROLLBACK-RI
TP-REPORT-RI
TP-INITIALIZE-RI
TP-INITIALIZE-RC
C-INITIALIZE-RI
C-INITIALIZE-RC
C-BEGIN-RI
C-BEGIN-RC
U-ASE APDU

NOTE 2 – The TP-INITIALIZE-RI/RC are the only TP APDUs mapped to the A-ASSOCIATE services. The U-ASE is free to map its APDUs on A-ASSOCIATE services.

NOTE 3 – Some of the TP APDUs in Table 46 are not (expected to be) concatenated because they are defined to be mapped into other service in isolation rather than because of any concatentation rule (e.g. TP-TOKEN-GIVE/PLEASE-RI is only ever carried on P-TOKEN-GIVE/PLEASE-RI in isolation).

f) The C-COMMIT-RI APDU can only be in a concatenation sequence containing itself and a following C-BEGIN-RI APDU. This concatenation sequence is only used on a *dialogue which is chaining*.

10.8 Routeing

The router part of the SACF receives CCR, ACSE and Presentation indications and confirms and determines which part of the PM shall receive the service primitive. This ensures that each service primitive is seen by the MACF (and possibly the SACF procedures) only once and that the full combined semantics of any TP APDU carried on the user data of a service primitive is given to the MACF with a single indication or confirm.

If the user data parameter associated with the service primitive is not present, the service primitive will be given to the MACF through the appropriate SACF procedures.

11 MACF description

11.1 Introduction

This clause describes the PM MACF procedures related to the use of TP Service primitives by the TPSUI.

These procedures identify which actions are to be taken by the MACF. Main procedures and Internal Event procedures that pertain to both the TPPM and the CPM, and that pertain only to the CPM, are identified accordingly in the subclause title. Subclause titles without such identification pertain only to the TPPM.

Dashed lists are used exclusively to present the actions.

NOTE – This representation makes it easier for the reader to locate the actions when using the "Index of Actions and Events" which follows clause 15.

11.2 CAF service definition

11.2.1 CAF-PLEASE request

This service is used by the TPPM to request the assignment of a channel for the purpose of initiating recovery. The issuance of the CAF-PLEASE request always results in a subsequent CAF-GIVE indication or CAF-FAIL indication, provided the TPPM still exists.

This service does not relate to any particular channel.

This is an unconfirmed service.

Table 47 gives the parameters of this primitive.

Table 47/X.862 – CAF-PLEASE parameters

CAF-PLEASE	
Parameter	**req**
AE-Title	M
Atomic Action Identifier	M
Branch Identifier	M
Superior	M

a) **AE-Title**: Specifies the Application Entity where a TPPM with the Atomic Action Identifier and Branch Identifier (as given in the following parameters) may be located.

NOTE – The sought TPPM may have completed commitment or rollback and ceased to exist.

b) **Atomic Action Identifier**: Together with the Branch Identifier, specifies the transaction branch to be recovered. Its value can be in the range as specified in ITU-T Rec. X.852 | ISO/IEC 9805-1.

c) **Branch Identifier**: Together with the Atomic Action Identifier, specifies the transaction branch to be recovered. Its value can be in the range as specified in ITU-T Rec. X.852 | ISO/IEC 9805-1.

d) **Superior**: Specifies whether the channel is for recovery of the branch to the superior. This can take the value:

1) "false", when the channel is for recovery of a branch to a subordinate;

2) "true", when the channel is for recovery of the branch to the superior.

The parameter values for "Atomic Action Identifier", "Branch Identifier", and "Superior" are obtained from the transaction and branch details in log-ready and log-commit records for the transaction branch being recovered. These parameters are not set explicitly in the procedures that follow, but are assumed to be set on each CAF-PLEASE request.

11.2.2 CAF-GIVE indication

This service indicates that a *channel has been transferred* to the TPPM for the purpose of recovery. This *transfer of the channel* is made because of a previous CAF-PLEASE request.

Upon issuance of this service primitive, the channel has been completely established and a C-RECOVER request may be immediately issued by the MACF.

NOTE – The SACF may queue the C-RECOVER request if the *token* is not owned.

This service relates to the *channel which is to be transferred*.

This is an unconfirmed service.

Table 48 gives the parameters of this primitive.

Table 48/X.862 – CAF-GIVE parameters

CAF-GIVE	
Parameter	**ind**
Channel-Utilization	M

– **Channel-Utilization**: Specifies the type of recovery to be performed on this channel. The values are:

1) "one-way-recovery";
2) "two-way-recovery".

11.2.3 CAF-FAIL indication

This service indicates that a request for the assignment of a channel for recovery cannot be satisfied by the CPM. This indication is a response to a previous CAF-PLEASE request.

The TPPM is responsible for issuing a further CAF-PLEASE request if recovery is still required.

This service does not relate to any particular channel.

This is an unconfirmed service.

This service has no parameters.

11.2.4 CAF-DETACH request

This service indicates that the TPPM has no further use for the channel. Issuing a CAF-DETACH request results in the *transfer of the channel* to the CPM.

This service relates to the *channel which is to be transferred*.

This is an unconfirmed service.

Table 49 gives the parameters of this primitive.

Table 49/X.862 – CAF-DETACH parameters

CAF-DETACH	
Parameter	**req**
Type	M

– **Type**: This parameter can take the value:

1) "clean-up", when the channel is being *detached* from the TPPM and a C-RECOVER indication or confirm may be subsequently received by the CPM;
2) "free", when the channel is available for another recovery exchange;
3) "not-used", when the TPPM did not use the channel and is returning it immediately after the issuance of the CAF-GIVE indication.

When this service primitive is referenced in the procedure clauses, the value listed in the parenthetical argument is the value of the Type parameter.

11.2.5 CAF-RECOVER indication

This service indicates that a *channel has been transferred* to the TPPM for the purpose of recovering a specific transaction branch. The indication identifies the transaction branch.

The channel has been completely established and the response to the recovery action can be sent immediately on the channel.

The service relates to a specific transaction branch. After the recovery action has concluded, the channel must be *detached* from the TPPM.

When this service primitive is referenced in the procedure clauses, the first value listed in the parenthetical argument is the value of the Recovery State parameter.

Table 50 gives the parameters of this primitive.

Table 50/X.862 – CAF-RECOVER parameters

CAF-RECOVER	
Parameter	**ind**
Recovery State	M
Atomic Action Identifier	M
Branch Identifier	M
Channel-Utilization	M
Heuristic-Report	C

a) **Recovery State**: Specifies the peer's view of the state of the branch. The values are:

   1) "ready";

   2) "commit".

b) **Atomic Action Identifier**: Together with the Branch Identifier, specifies the transaction branch to be recovered. Its value can be in the range as specified in ITU-T Rec. X.852 | ISO/IEC 9805-1.

c) **Branch Identifier**: Together with the Atomic Action Identifier, specifies the transaction branch to be recovered. Its value can be in the range as specified in ITU-T Rec. X.852 | ISO/IEC 9805-1.

d) **Channel-Utilization**: Specifies the type of recovery to be performed on this channel. The values are:

   1) "one-way-recovery";

   2) "two-way-recovery".

e) **Heuristic-Report**: The parameter is as described in the "Primitives and parameters" subclause for the TP-HEURISTIC-REPORT service in ITU-T Rec. X.861 | ISO/IEC 10026-2. It is absent unless the Dynamic Commit functional unit was selected on the original dialogue.

11.3 Main procedures

These procedures are invoked by TP, AF, SAF, CAF, and CCR services.

11.3.1 TP-BEGIN-DIALOGUE request

Depending on a local decision, either:

– assign an association compatible with this dialogue;

  NOTE 1 – Whether this association is assigned from the pool of associations, or has been newly established for use with this dialogue is a local matter.

  NOTE 2 – If the TP-BEGIN-DIALOGUE request is being issued in response to a dialogue solicitation, the association assigned will be the one the solicitation arrived on.

– issue an AF-BEGIN-DIALOGUE request;

– invoke the "Initiating a transaction branch" procedure (see 11.5.9), if dialogue is coordinated; or

– issue a TP-BEGIN-DIALOGUE (rejected(provider)) confirm with the Rollback parameter set to "false" and the Diagnostic parameter set to:

  a) "recipient-unknown", if this dialogue establishment request is being rejected as a result of an A-ASSOCIATE confirm with the Diagnostic parameter set to one of:

     1) "called AP title not recognized";

2) "called AE qualifier not recognized";

3) "called AP invocation-identifier not recognized";

4) "called AE invocation-identifier not recognized".

b) Optionally, "functional-unit-not-supported" if the TPPM is unable to obtain an association compatible with the Functional Units selected.

c) "no-reason-given" otherwise.

NOTE 3 – The TPPM will normally only choose to issue the TP-BEGIN-DIALOGUE (rejected) confirm because it is unable to obtain a compatible association, either from the pool, or by establishing a new association. If a TP-BEGIN-DIALOGUE (rejected) confirm is issued because of a failed attempt to establish an association, it is a local matter to convey the parameter from the A-ASSOCIATE confirm (other than those enumerated above) to the TPSUI.

11.3.2 AF-BEGIN-DIALOGUE indication (TPPM and CPM)

If the dialogue or channel is not to be rejected:

a) If a dialogue, and the *dialogue will not be coordinated*:

– Create a TPSUI of the type specified by the Recipient-TPSU-Title parameter of the AF-BEGIN-DIALOGUE indication, or a TPSUI of a default type if the parameter is not present.

NOTE 1 – From the OSIE perspective, whether a new TPSUI is "created" in a real open system or whether an old instance is re-used, is a local matter.

– Issue a TP-BEGIN-DIALOGUE indication.

b) If a dialogue, and the *dialogue will be coordinated*:

– continue.

NOTE 2 – If the *dialogue will be coordinated*, a TP-BEGIN-DIALOGUE indication will be issued on receipt of a C-BEGIN indication (see 11.3.37).

c) If a channel:

– issue an AF-BEGIN-DIALOGUE (accepted, dataRI) response.

If the dialogue or channel is to be rejected:

– Issue an AF-BEGIN-DIALOGUE (rejected(provider), dataRI) response with the value of the:

a) Diagnostic parameter, if a dialogue, set to, as appropriate, one of:

1) "recipient-tpsu-title-unknown";

2) "tpsu-not-available(permanent)";

3) "tpsu-not-available(transient)";

4) "recipient-tpsu-title-required";

5) "functional-unit-not-supported";

6) "functional-unit-combination-not-supported";

7) "subordinate-shall-be-commit-superior";

8) "superior-shall-be-commit-subordinate";

9) "FU-not-acceptable";

10) "no-reason-given".

b) Diagnostic parameter, if a channel, set to, as appropriate, one of:

1) "functional-unit-not-supported";

2) "tppm-recovery-not-available";

3) "two-way-recovery-not-supported";

4) "no-reason-given".

c) Functional-Units parameter set to the functional units which are supported, if the Diagnostic parameter is set to "functional-unit-not-supported".

– Issue an SAF-DETACH-ASSOCIATION (free) request if the Commit functional unit is not selected, or if a channel.

– Issue an SAF-DETACH-ASSOCIATION (begin-fear) request if the Unchained Transactions functional unit is selected and the *dialogue will be coordinated*.

– Issue an SAF-DETACH-ASSOCIATION (begin-indication-expected) request if the *dialogue will be coordinated.*

11.3.3 TP-BEGIN-DIALOGUE response

If the dialogue is not coordinated and the Result parameter is:

a) "accepted":

  – invoke the "First request/response" procedure (see 11.5.8).

b) "rejected":

  – issue an AF-BEGIN-DIALOGUE (rejected(user), dataRI) response;

  – issue an SAF-DETACH-ASSOCIATION (begin-fear) request if the Unchained Transactions functional unit is selected;

  – issue an SAF-DETACH-ASSOCIATION (free) request if the Unchained Transactions functional unit is not selected.

If the dialogue is coordinated and the Result parameter is:

a) "accepted", and the TPPM is in the:

  1) ACTIVE state:

    – invoke the "First request/response" procedure (see 11.5.8).

  2) DECIDED (rollback) state:

    – continue.

    NOTE 1 – This situation occurs when the TPPM received the C-ROLLBACK indication before the TPSUI has issued TP-BEGIN-DIALOGUE (accepted) response. The TP-BEGIN-DIALOGUE-RC APDU will be sent as User Data of the *rollback response*, which will be issued upon the receipt of a TP-DONE request.

b) "rejected", and the TPPM is in the:

  1) ACTIVE state:

    – issue an AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) response;

    – issue an SAF-DETACH-ASSOCIATION (rollback-confirm-expected) request;

    – cease to be part of the transaction.

  2) DECIDED (rollback) state:

    NOTE 2 – The only time the TPPM could be in this state is if it is a leaf and it received a *rollback indication* from the superior.

    – issue an AF-BEGIN-DIALOGUE (rejected(user), rollbackRC) response;

    – issue an SAF-DETACH-ASSOCIATION (free) request;

    – cease to be part of the transaction.

11.3.4 AF-BEGIN-DIALOGUE (accepted) confirm on a Dialogue

If the value of the Mapping parameter is "rollbackRC", the TPPM shall be in the DECIDED (rollback) state.

If the value of the Mapping parameter is "dataRI" or "rollbackRC", do the following:

a) If the Confirmation parameter of the TP-BEGIN-DIALOGUE request was "always" and a TP-U-ABORT request was not received:

  – issue a TP-BEGIN-DIALOGUE confirm with the rollback parameter set to "false".

b) If the Confirmation parameter of the TP-BEGIN-DIALOGUE request was "always" and the Mapping parameter is "dataRI" and a TP-U-ABORT request was received:

  – continue.

c) If the Confirmation parameter of the TP-BEGIN-DIALOGUE request was "negative":

   – continue.

d) If the value of the Mapping parameter is "rollbackRC":

   1) If a TP-U-ABORT request was received and no AF-ABORT request was issued:

      – issue an AF-ABORT (user, dataRI) request.

        NOTE – If a TP-U-ABORT request was received, an AF-ABORT request could not have been issued if the Mapping parameter is "rollbackRC".

      – Issue an SAF-DETACH-ASSOCIATION (free) request.

   2) If the *last rollback confirm was received* and this is an intermediate node and the superior *dialogue has not been detached*:

      – invoke the "Reporting rollback to superior" procedure (see 11.5.16).

   3) If the *last rollback confirm was received* and the *superior dialogue is not chaining* and *rollback reporting has completed*:

      – invoke the "Initiating transaction after rollback" procedure (see 11.5.11).

   4) If the Unchained Transactions functional unit is selected on the dialogue and no TP-U-ABORT request was received:

      – *close the PSAP*.

11.3.5 AF-BEGIN-DIALOGUE (rejected, dataRI) confirm on a Dialogue

If the Diagnostic parameter is set to "association-reserved", set the value of that parameter for the TP-BEGIN-DIALOGUE confirm to "no-reason-given".

If the dialogue is not coordinated:

– issue a TP-BEGIN-DIALOGUE confirm with the value of the Rollback parameter set to "false";

– issue an SAF-DETACH-ASSOCIATION (free) request.

If the dialogue is coordinated and either:

a) the Unchained Transactions functional unit is selected and the Type parameter is "rejected(user)"; or

b) the Type parameter is "rejected(provider)";

and the TPPM is in the:

a) ACTIVE state and a *transaction completion request* was received or in the READY state or in the ONE-PHASE state:

   – issue a TP-BEGIN-DIALOGUE confirm with the Rollback parameter set to "true";

   – issue an SAF-DETACH-ASSOCIATION (rollback-indication-expected) request;

   – invoke the "Initiating rollback at TPPM" procedure (see 11.5.10), with a diagnostic-value of "dialogue-reject-transaction-completion-collision.

b) ACTIVE state and a *transaction completion request* was not received:

   – issue a TP-BEGIN-DIALOGUE confirm with the Rollback parameter set to "false";

   – issue an SAF-DETACH-ASSOCIATION (rollback-indication-expected) request.

c) DECIDED (rollback) state:

   1) Always:

      – issue an SAF-DETACH-ASSOCIATION (rollback-confirm-expected) request.

   2) If a TP-U-ABORT request was not received:

      – issue a TP-BEGIN-DIALOGUE confirm with the Rollback parameter set to "false".

        NOTE – *A TP-DONE request is now owed.*

3) If a TP-U-ABORT request was received and the *last rollback confirm was received*:

   i) If this is an intermediate node and the superior *dialogue has not been detached*:

      – invoke the "Reporting rollback o superior" procedure (see 11.5.16).

   ii) If the superior *dialogue is not chaining* and *rollback reporting has completed*:

      – invoke the "Initiating transaction after rollback" procedure (see 11.5.11).

11.3.6 AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) confirm

If the dialogue is coordinated and if the TPPM is in the:

a) ACTIVE state and a *transaction completion request* was received or in the READY state or in the ONE-PHASE state:

   – issue a TP-BEGIN-DIALOGUE confirm with the Rollback parameter set to "true";

   – invoke the "Initiating rollback at TPPM" procedure (see 11.5.10), with a diagnostic-value of "dialogue-reject-transaction-completion-collision;

   – issue an SAF-DETACH-ASSOCIATION (free) request.

b) ACTIVE state and a TP-COMMIT request was not received:

   – issue a TP-BEGIN-DIALOGUE confirm with the Rollback parameter set to "false";

   – issue a C-ROLLBACK response;

   – issue an SAF-DETACH-ASSOCIATION (free) request.

c) DECIDED (rollback) state:

   1) Always:

      – issue a C-ROLLBACK response;

      – issue an SAF-DETACH-ASSOCIATION (free) request.

   2) If a TP-U-ABORT request was not received:

      – issue a TP-BEGIN-DIALOGUE confirm with the Rollback parameter set to "false".

      NOTE – *A TP-DONE request is now owed.*

   3) If a TP-U-ABORT request was received and the *last rollback confirm was received*:

      i) If this is an intermediate node and the superior *dialogue has not been detached*:

         – invoke the "Reporting rollback to superior" procedure (see 11.5.16).

      ii) If the superior *dialogue is not chaining* and *rollback reporting has completed*:

         – invoke the "Initiating transaction after rollback" procedure (see 11.5.11).

11.3.7 AF-BEGIN-DIALOGUE (rejected(user), rollbackRC) confirm

If dialogue is coordinated, and the TPPM is in the DECIDED (rollback) state:

a) Always:

   – issue an SAF-DETACH-ASSOCIATION (free) request.

b) If a TP-U-ABORT request was not received:

   – issue a TP-BEGIN-DIALOGUE confirm with the Rollback parameter set to "false".

   NOTE – *A TP-DONE request is now owed.*

c) If a TP-U-ABORT request was received and the *last rollback confirm was received*:

   1) If this is an intermediate node and the superior *dialogue has not been detached*:

      – invoke the "Reporting rollback to superior" procedure (see 11.5.16).

   2) If the superior *dialogue is not chaining* and *rollback reporting has completed*:

      – invoke the "Initiating transaction after rollback" procedure (see 11.5.11).

11.3.8 AF-BEGIN-DIALOGUE confirm (CPM)

If the Result parameter is "accepted" and a *CAF-PLEASE request is outstanding* for the channel, and a TPPM with a branch corresponding to the "Atomic Action Identifier", "Branch Identifier", and "Superior" parameters specified in the previous CAF-PLEASE request is found:

– *transfer the channel* to the requesting TPPM;

– issue a CAF-GIVE indication to the requesting TPPM.

If the Result parameter is "accepted" and either *a CAF-PLEASE request is not outstanding* for the channel or a TPPM with a branch corresponding to the "Atomic Action Identifier", "Branch Identifier", and "Superior" parameters specified in the previous CAF-PLEASE request is not found:

– retain control of the channel.

NOTE – The CPM may make local decisions to allocate the channel to another TPPM in response to a subsequent CAF-PLEASE request, or to terminate the channel. If the channel supports two-way-recovery, it will pass control of the channel to its peer in response to a subsequent AF-TOKEN-PLEASE indication.

If the Result parameter is "rejected(provider)":

– invoke the "Fail an outstanding CAF-PLEASE request" procedure (see 11.5.7), if a *CAF-PLEASE request is outstanding* for the channel;

– issue an SAF-DETACH-ASSOCIATION (free) request.

11.3.9 SAF-ASSOCIATION-LOST indication

NOTE 1 – It is a local matter whether an implementation retries establishing a dialogue on another association.

If the dialogue is not coordinated:

– issue a TP-BEGIN-DIALOGUE (rejected(provider)) confirm with the value of the Rollback parameter set to "false" and with the Diagnostic parameter set to "no-reason-given".

If dialogue is coordinated and if the TPPM is in the:

a) ACTIVE state and a *transaction completion request* was received or in the READY state or in the ONE-PHASE state:

   – issue a TP-BEGIN-DIALOGUE (rejected(provider)) confirm with the Rollback parameter set to "true" and with the Diagnostic parameter set to "no-reason-given";

   – invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "local-rollback".

b) ACTIVE state and a *transaction completion request* was not received:

   – issue a TP-BEGIN-DIALOGUE (rejected(provider)) confirm with the Rollback parameter set to "false" and with the Diagnostic parameter set to "no-reason-given".

c) DECIDED (rollback) state:

   1) If a TP-U-ABORT request was not received:

      – issue a TP-BEGIN-DIALOGUE (rejected(provider)) confirm with the Rollback parameter set to "false" and with the Diagnostic parameter set to "no-reason-given".

      NOTE 2 – *A TP-DONE request is now owed.*

   2) If a TP-U-ABORT request was received and the *last rollback confirm was received*:

      i) if this is an intermediate node and the superior *dialogue has not been detached*:

         – invoke the "Reporting rollback to superior" procedure (see 11.5.16);

      ii) if the superior *dialogue is not chaining* and *rollback reporting has completed*:

         – invoke the "Initiating transaction after rollback" procedure (see 11.5.11).

11.3.10 SAF-ASSOCIATION-LOST indication (CPM)

If a *CAF-PLEASE request is outstanding* for the channel:

– invoke the "Fail an outstanding CAF-PLEASE" procedure (see 11.5.7);

otherwise:

– continue.

11.3.11 TP-END-DIALOGUE request

- Invoke the "First request/response" procedure (see 11.5.8), if this is a superior dialogue and if no AF-BEGIN-DIALOGUE response has been issued on the dialogue since an AF-BEGIN-DIALOGUE indication has been received;
- issue an AF-END-DIALOGUE request.

If the value of the Confirmation parameter is "false":

- issue an SAF-DETACH-ASSOCIATION (begin-fear) request if the Unchained Transactions functional unit is selected and this is a superior dialogue;
- issue an SAF-DETACH-ASSOCIATION (free) request if the Unchained Transactions functional unit is not selected or this is a subordinate dialogue.

11.3.12 AF-END-DIALOGUE indication

If the dialogue is not coordinated, and:

a) The value of the Confirmation parameter is "false":
   - issue a TP-END-DIALOGUE indication;
   - issue an SAF-DETACH-ASSOCIATION (free) request.

b) The value of the Confirmation parameter is "true" and the dialogue is in a *user error purging period*:
   1) if the Polarized Control functional unit is selected:
      - issue a TP-END-DIALOGUE indication;
   2) if the Shared Control functional unit is selected:
      - continue.

c) The value of the Confirmation parameter is "true", the dialogue is not in a *user error purging period*, and:
   1) a dialogue termination request is outstanding and the Shared Control functional unit is selected:
      - issue a TP-P-ABORT indication with the value of the Diagnostic parameter set to "end-dialogue-collision" and the value of the Rollback parameter set to "false";
      - issue an SAF-DETACH-ASSOCIATION (free) request;
   2) a dialogue termination request is not outstanding:
      - issue a TP-END-DIALOGUE indication.

If the dialogue is coordinated, the indication was from a subordinate, a TP-BEGIN-TRANSACTION request has been received, a C-BEGIN confirm was not received, and the TPPM is in the:

a) ACTIVE state, a *transaction completion request* was not received, and the Confirmation parameter is set to "false":
   - issue a TP-P-ABORT indication with the Diagnostic parameter set to "begin-transaction-end-dialogue-collision" and with the Rollback parameter set to "false";
   - issue an SAF-DETACH-ASSOCIATION (rollback-indication-expected) request.

b) ACTIVE state, a *transaction completion request* was received, and the Confirmation parameter is set to "false":
   - issue a TP-P-ABORT indication with the Diagnostic parameter set to "begin-transaction-end-dialogue-collision" and with the Rollback parameter set to "true";
   - issue an SAF-DETACH-ASSOCIATION (rollback-indication-expected) request;
   - invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "end-dialogue-transaction-completion-collision".

c) ACTIVE state, the Confirmation parameter is set to "true", and the dialogue is in the *transaction initiation purging period*:
   - continue.

d) ACTIVE state, a *transaction completion request* was not received, the Confirmation parameter is set to "true", and the dialogue is not in the *transaction initiation purging period*:

- issue a TP-P-ABORT indication with the Diagnostic parameter set to "begin-transaction-end-dialogue-collision" and with the Rollback parameter set to "false";
- issue an SAF-DETACH-ASSOCIATION (rollback-indication-expected) request.

e) ACTIVE state, a *transaction completion request* was received, the Confirmation parameter is set to "true", and the dialogue is not in the *transaction initiation purging period*:

- issue a TP-P-ABORT indication with the Diagnostic parameter set to "begin-transaction-end-dialogue-collision" and with the Rollback parameter set to "true";
- issue an SAF-DETACH-ASSOCIATION (rollback-indication-expected) request;
- invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "end-dialogue-transaction-completion-collision".

f) DECIDED (rollback) state and the dialogue is not in the *transaction initiation purging period*:

1) Always:

   - issue an SAF-DETACH-ASSOCIATION (rollback-confirm-expected) request.

2) If a TP-U-ABORT request was not received:

   - issue a TP-P-ABORT indication with the Diagnostic parameter set to "begin-transaction-end-dialogue-collision" and with the Rollback parameter set to "false".

   NOTE – *A TP-DONE request is now owed.*

3) If a TP-U-ABORT request was received and the *last rollback confirm was received*:

   i) if this is an intermediate node and the superior *dialogue has not been detached*:

      - invoke the "Reporting rollback to superior" procedure (see 11.5.16);

   ii) if the superior *dialogue is not chaining* and *rollback reporting has completed*:

      - invoke the "Initiating transaction after rollback" procedure (see 11.5.11).

g) DECIDED (rollback) state and the dialogue is in the *transaction initiation purging period*:

- continue.

h) READY or ONE-PHASE state and either the Confirmation parameter is set to "false" or the dialogue is not in the *transaction initiation purging period*:

- issue a TP-P-ABORT indication with the Diagnostic parameter set to "begin-transaction-end-dialogue-collision" and with the Rollback parameter set to "true";
- issue an SAF-DETACH-ASSOCIATION (rollback-indication-expected) request;
- invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "end-dialogue-transaction-completion-collision".

i) READY or ONE-PHASE state, the Confirmation parameter is set to "true", and the dialogue is in the *transaction initiation purging period*:

- continue.

11.3.13 AF-END-DIALOGUE indication (CPM)

- Invoke the "Fail an outstanding CAF-PLEASE request" procedure (see 11.5.7), if a *CAF-PLEASE request is outstanding* for the channel;
- issue an SAF-DETACH-ASSOCIATION (free) request.

11.3.14 TP-END-DIALOGUE response

- Invoke the "First request/response" procedure (see 11.5.8), if this is a superior dialogue and if no AF-BEGIN-DIALOGUE response has been issued on the dialogue since an AF-BEGIN-DIALOGUE indication has been received;
- issue an AF-END-DIALOGUE response;
- issue an SAF-DETACH-ASSOCIATION (free) request.

11.3.15 AF-END-DIALOGUE confirm

If the Confirmation parameter of the TP-END-DIALOGUE request previously issued was "true":

- issue a TP-END-DIALOGUE confirm;
- issue an SAF-DETACH-ASSOCIATION (free) request.

11.3.16 TP-U-ERROR request

- Invoke the "First request/response" procedure (see 11.5.8), if this is a superior dialogue and if no AF-BEGIN-DIALOGUE response has been issued on the dialogue since an AF-BEGIN-DIALOGUE indication has been received;
- issue an AF-U-ERROR request.

11.3.17 AF-U-ERROR indication

If the TPPM is in the ACTIVE state and already issued an AF-PREPARE request on this dialogue or is in the ACTIVE state and a *transaction completion request* has been received:

- invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic value of "user-data-transaction-completion-collision".

If the TPPM is in the ACTIVE state and no AF-PREPARE request has been issued and no *transaction completion request* has been received, or if the coordination level is "none":

a) If the Shared Control functional unit was selected:

   - issue a TP-U-ERROR indication;
   - issue an AF-U-ERROR response if both of the following conditions are met:

     1) this is a subordinate dialogue or an AF-BEGIN-DIALOGUE response was issued; and

        NOTE 1 – The AF-U-ERROR response will be issued after the AF-BEGIN-DIALOGUE response.

     2) there is no dialogue termination request outstanding and no handshake request outstanding.

b) If the Polarized Control functional unit was selected:

   - continue, if the TPPM is in a *user error purging period*;
   - issue a TP-U-ERROR indication if the TPPM is not in a *user error purging period*.

If the TPPM is in the DECIDED (rollback) state:

- continue.

If the TPPM is in the READY, ONE-PHASE or READ-ONLY state and a C-READY request *or ready-substitute request* was sent on this dialogue, no AF-PREPARE indication was received, the Shared Control Functional Unit is selected and at least one of the Dynamic Commit, Implicit Prepare or One-phase Commit functional units were selected:

- continue.

  NOTE 2 – This can only occur if either this or the peer TPSUI has issued primitives in violation of the application semantics. The peer TPPM will detect the error when it receives the C-READY indication (or *ready-substitute indication*).

11.3.18 AF-U-ERROR confirm

If the Shared Control functional unit is selected:

- continue.

11.3.19 TP-U-ABORT request

NOTE 1– There are some cases where an AF-ABORT request will not be issued by this procedure. If the transaction branch has not completed the termination phase, the issuance of the AF-ABORT request is deferred until the necessary response/request is issued to complete the transaction. This is required because, according to CCR rules, nothing can be issued until this time. If the AF-ABORT request is to be deferred, this procedure will either continue or initiate rollback.

- Invoke the "First request/response" procedure (see 11.5.8), if this is a superior dialogue and if no AF-BEGIN-DIALOGUE response has been issued on the dialogue since an AF-BEGIN-DIALOGUE indication has been received and the TPPM is not in the DECIDED (rollback) state.

If the coordination level is "none":

- issue an AF-ABORT (user, dataRI) request;
- issue an SAF-DETACH-ASSOCIATION (begin-fear) request if the Unchained Transactions functional unit is selected and this is a superior dialogue;
- issue an SAF-DETACH-ASSOCIATION (free) request if the Unchained Transactions functional unit is not selected or this is a subordinate dialogue.

If the dialogue is coordinated, and the TPPM is in the:

a) ACTIVE state and:

   1) A C-NOCHANGE (result-not-required) indication has been received:

      - issue an AF-ABORT (user, nochangeRC) request with the Outcome parameter set to "not-determined";
      - issue an SAF-DETACH-ASSOCIATION (free) request.

   2) A C-NOCHANGE (result-requested) indication or AF-NOCHANGE indication has been received:

      - continue.

   3) No C-NOCHANGE indication or AF-NOCHANGE indication has been received:

      - issue an AF-ABORT (user, rollbackRI) request if the dialogue is to a subordinate;
      - issue a C-CANCEL request if the dialogue is to the superior, the Cancel functional unit is selected and subject to a local decision;

        NOTE 2 – The TPPM does not issue a C-ROLLBACK request to the superior at this time. It must wait for TP-DONE and all *rollback confirms*.

      - invoke the "Initiating rollback at TPPM" procedure (see 11.5.10).

b) READY state:

   - continue.

c) DECIDED (commit) state, and:

   1) The TP-U-ABORT request pertains to the superior dialogue, and the TPPM:

      i) has received a C-COMMIT+C-BEGIN indication:

         - invoke the "Rollback next transaction" procedure (see 11.5.17);

      ii) has received a C-COMMIT indication on this dialogue and an *intermediate log-record has not been rewritten*:

         - continue;

      iii) has received a C-COMMIT indication on this dialogue and an *intermediate log-record has been rewritten*:

         - issue an AF-ABORT (user, dataRI) request;
         - issue an SAF-DETACH-ASSOCIATION (begin-fear) request;

      iv) has sent a C-COMMIT request on this superior dialogue and has received a *commit confirm*:

         - issue an AF-ABORT (user, dataRI) request;
         - issue an SAF-DETACH-ASSOCIATION (begin-fear) request;

      v) has sent a C-COMMIT request on this superior dialogue and has not received a *commit confirm*:

         - continue;

      vi) has received a C-NOCHANGE indication on this superior dialogue and *reporting does not apply on the dialogue*:

         - issue an AF-ABORT (user, dataRI) request;
         - issue an SAF-DETACH-ASSOCIATION (begin-fear) request;

vii) has received an AF-NOCHANGE indication on this superior dialogue and *reporting does not apply on the dialogue*:

- issue an AF-ABORT (user, dataRI) request;
- issue an SAF-DETACH-ASSOCIATION (begin-fear) request;
- invoke the "Rollback next transaction" procedure (see 11.5.17);

viii) has received a C-NOCHANGE indication on this superior dialogue and *reporting applies on the dialogue*:

- continue;

ix) has received an AF-NOCHANGE indication on this superior dialogue and *reporting applies on the dialogue*:

- invoke the "Rollback next transaction" procedure (see 11.5.17).

2) The TP-U-ABORT request pertains to a subordinate dialogue, and the TPPM:

i) has sent a C-COMMIT+C-BEGIN request:

- invoke the "Rollback next transaction" procedure (see 11.5.17);

ii) has sent a C-COMMIT request and has received a *commit confirm*:

- issue an AF-ABORT (user, dataRI) request;
- issue an SAF-DETACH-ASSOCIATION (free) request;

iii) has sent a C-COMMIT request and has not received a *commit confirm*:

- continue;

iv) has received a C-COMMIT indication and sent a C-COMMIT response:

- issue an AF-ABORT (user, dataRI) request;
- issue an SAF-DETACH-ASSOCIATION (free) request;

v) has a received a C-COMMIT indication and has not sent a C-COMMIT response:

- continue;

vi) has received a C-NOCHANGE indication and no C-NOCHANGE response has been issued:

- issue an AF-ABORT (user, nochangeRC) request with the Outcome parameter set to "not-determined";
- issue an SAF-DETACH-ASSOCIATION (free) request.

NOTE 3 – If a C-NOCHANGE response has been issued, the dialogue will no longer be coordinated

d) DECIDED (rollback) state and a *rollback response* was issued or a *rollback confirm* was received and:

1) The TP-U-ABORT request pertains to the superior dialogue and:

i) a C-BEGIN indication was not received while in the DECIDED (rollback) state:

- issue an AF-ABORT (user, dataRI) request;
- *open the PSAP* if the Unchained Transactions functional unit is selected;
- issue an SAF-DETACH-ASSOCIATION (begin-fear) request;

NOTE 4 – *A TP-DONE request is owed*, which will invoke the "Initiating Next Transaction after Rollback" procedure.

ii) a C-BEGIN indication was received while in the DECIDED (rollback) state:

- issue an AF-ABORT (user, rollbackRI) request on the superior dialogue;
- *open the PSAP*;
- issue an SAF-DETACH-ASSOCIATION (rollback-confirm-expected) request.

2) The TP-U-ABORT request pertains to a subordinate dialogue:

- issue an AF-ABORT (user, dataRI) request;
- *open the PSAP*, if the Unchained Transactions functional unit is selected;
- issue an SAF-DETACH-ASSOCIATION (free) request.

e) DECIDED (rollback) state and neither a *rollback response* was issued nor a *rollback confirm* was received:

   – continue.

f) DECIDED (commit-one-phase) state or DECIDED (unknown) state or READ-ONLY or EARLY-EXIT state and a TP-UNKNOWN indication was issued and:

   1) The TP-U-ABORT request pertains to the superior dialogue and:

      i) a C-BEGIN indication was not received while in the DECIDED or READ-ONLY or EARLY-EXIT state:

         – issue an AF-ABORT (user, dataRI) request;

         – *open the PSAP* if the Unchained Transactions functional unit is selected;

         – issue an SAF-DETACH-ASSOCIATION (begin-fear) request;

         NOTE 5 – *A TP-DONE request is owed*, which will invoke the "Completing ONE-PHASE or READ-ONLY" procedure.

      ii) a C-BEGIN indication was received while in the ONE-PHASE or READ-ONLY or EARLY-EXIT state:

         – issue an AF-ABORT (user, rollbackRI) request on the superior dialogue;

         – *open the PSAP*;

         – issue an SAF-DETACH-ASSOCIATION (rollback-confirm-expected) request;

   2) The TP-U-ABORT request pertains to a subordinate dialogue and:

      i) a C-NOCHANGE confirm was received and *reporting does not apply to the dialogue*:

         – issue an AF-ABORT (user, dataRI) request;

         – *open the PSAP*;

         – issue an SAF-DETACH-ASSOCIATION (free) request;

      ii) a C-NOCHANGE or AF-NOCHANGE request was issued and *reporting applies to the dialogue*:

         – continue;

      iii) a C-NOCHANGE indication was received and no C-NOCHANGE response has been issued:

         – issue an AF-ABORT (user, nochangeRC) request with the Outcome parameter set to "not-determined";

         – issue an SAF-DETACH-ASSOCIATION (free) request;

      iv) an AF-EARLY-EXIT indication was received:

         – issue an AF-ABORT (user, dataRI) request;

         – issue an SAF-DETACH-ASSOCIATION (free) request.

g) ONE-PHASE state or READ-ONLY state or EARLY-EXIT state and neither a TP-COMMIT indication nor a TP-UNKNOWN indication was issued:

   – continue.

11.3.20 AF-ABORT (user, dataRI) indication

If the dialogue is not coordinated:

– issue a TP-U-ABORT indication with the Rollback parameter set to "false";

– issue an SAF-DETACH-ASSOCIATION (free) request.

If the dialogue is coordinated, the indication was from a subordinate, and the TPPM is in the:

a) ACTIVE state, and:

   1) a *transaction completion request* was not received;

   2) a C-BEGIN confirm was not received,

   then:

   – issue a TP-U-ABORT indication with the Rollback parameter set to "false";

   – issue an SAF-DETACH-ASSOCIATION (rollback-indication-expected) request.

b) ACTIVE state, and:

   1) a *transaction completion request* was received;

   2) a C-BEGIN confirm was not received,

   then:

   – issue a TP-U-ABORT indication with the Rollback parameter set to "true";

   – issue an SAF-DETACH-ASSOCIATION (rollback-indication-expected) request;

   – invoke the "Initiating rollback at TPPM" procedure (see 11.5.10).

c) DECIDED (rollback) state, and:

   1) if a *rollback confirm* was received or a *rollback response* was issued, and the *dialogue is chaining*:

      – issue a TP-U-ABORT indication with the Rollback parameter set to "false";

      – issue an SAF-DETACH-ASSOCIATION (free) request;

      NOTE 1 – In this situation, a TP-U-ABORT request cannot have been previously received on this association because, in that case, the association would have been immediately or eventually *detached* and the event causing this procedure invocation could not have happened.

   2) if a *rollback confirm* was not received and the Unchained Transactions functional unit is selected:

      i) if a TP-U-ABORT request was not received:

         – issue a TP-U-ABORT indication with the Rollback parameter set to "false";

      ii) always:

         – issue an SAF-DETACH-ASSOCIATION (rollback-confirm-expected) request;

      iii) if the *last rollback confirm was received* and this is an intermediate node and the superior *dialogue has not been detached*:

         – invoke the "Reporting rollback to superior" procedure (see 11.5.16);

      iv) if the *last rollback confirm was received*, the superior *dialogue is not chaining*, and *rollback reporting has completed*:

         – invoke the "Initiating transaction after rollback" procedure (see 11.5.11).

d) DECIDED (commit-one-phase) or DECIDED (unknown) state:

   – issue a TP-U-ABORT indication with the Rollback parameter set to "false", if a TP-U-ABORT request was not received;

   – issue an SAF-DETACH-ASSOCIATION (free) request.

If the dialogue is coordinated and the indication is from the superior, and the TPPM is in the DECIDED (rollback) state and has issued a *rollback response* to or received a *rollback confirm* from the superior:

– issue a TP-U-ABORT indication with the Rollback parameter set to "false";

– issue an SAF-DETACH-ASSOCIATION (free) request.

NOTE 2 – In this situation, a TP-U-ABORT request could not have been previously received on this dialogue because the association would have been *detached* since the rollback report to the superior occurred (and then the TPPM would not have received the AF-ABORT indication). Therefore, *a TP-DONE request must be owed*, preventing the transaction after the rollback from beginning. In addition, the *superior dialogue is* necessarily *chaining* and a C-BEGIN indication could not have been received in this state, since this would have led to *closing the PSAP*.

If the dialogue is coordinated and the indication is from the superior, and the TPPM is in the DECIDED (commit-one-phase) state:

– issue a TP-U-ABORT indication with the Rollback parameter set to "false";

– issue an SAF-DETACH-ASSOCIATION (free) request.

NOTE 3 – The dialogue is chaining and the TPPM is waiting for the C-BEGIN indication. A TP-U-ABORT request could not have been previously received on this dialogue because the association would have been *detached* and the TPPM would not have received the AF-ABORT indication.

If this is the superior dialogue and has a coordination level of "one-phase commitment", the Chained functional unit is selected on the dialogue and the TPPM is in the DECIDED (commit) state:

– invoke the "Rollback next transaction" procedure (see 11.5.17);

– issue a TP-U-ABORT indication with the Rollback parameter set to "false", if a TP-U-ABORT request was not received;

– issue an SAF-DETACH-ASSOCIATION (free) request.

11.3.21 Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm on a dialogue

NOTE – The A-ABORT request in the title of this procedure refer to when these service are issued by the U-ASE or some part of the SACF to ACSE. The A-ABORT indication includes the case where the service primitive has the value of the Abort Source parameter equal to "ACSE service-user", and is therefore indicated by ACSE to the U-ASE.

If this is a protocol error or internal error:

– issue an AF-ABORT (provider, abortRI) request with the Diagnostic parameter set to:

  1) "protocol error", if this is a protocol error;

  2) "transient-failure" or "permanent-failure", based on a local decision, if this is an internal error.

If no TP-U-ABORT request was received:

– issue a TP-P-ABORT indication with the Rollback parameter set to:

  a) "true", if the TPPM is in the ACTIVE state, does not have a dialogue establishment indication outstanding, the dialogue is coordinated and no C-NOCHANGE (not-required) indication and no AF-EARLY-EXIT indication has been received;

  b) "false", otherwise;

  and the Diagnostic parameter set to:

  a) "protocol-error", if this is a protocol error;

  b) "transient-failure" or "permanent-failure", based on a local decision, if this is an internal error;

  c) "permanent-failure", if this is an A-RELEASE response, an A-RELEASE confirm, an A-[P-] ABORT indication, or an A-ABORT request; or

  d) to the Diagnostic parameter on the AF-ABORT indication, if this is an AF-ABORT indication;

– cease to be part of the transaction, if there is a dialogue establishment indication outstanding.

If the TPPM is in the ACTIVE or DECIDED (rollback) state and all of the following are true:

1) the dialogue is to a subordinate and is coordinated;

2) an AF-PREPARE request has been sent to this subordinate or the Implicit Prepare functional unit is selected on the dialogue*;*

3) *heuristic reporting applies* to this dialogue;

4) no *ready-substitute indication* has been received from this subordinate; and

5) no *rollback confirm* or *rollback indication* has been received from this subordinate,

then:

– issue a TP-HEURISTIC-REPORT indication with the Heuristic-Report parameter set to "heuristic-hazard";

– *write a log-damage record* with the value "heuristic-hazard" if no log-damage record exists.

If the dialogue is coordinated and no *ready-substitute indication has been received* on the dialogue and the TPPM is in the:

a) ACTIVE state and the dialogue is with a subordinate or is with the superior and the TPPM does not have a dialogue establishment indication outstanding:

  - invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "local-rollback".

b) READY state and:

  i) a *ready signal has been sent* on the dialogue:

    - issue a CAF-PLEASE request with the AE-Title set to the value of the AE-Title taken from the branch identifier for this neighbour contained in the log-ready record;

  ii) no *ready signal has been sent* on the dialogue:

    - continue.

c) DECIDED (commit) state and:

  i) a C-COMMIT+C-BEGIN indication was received or a C-COMMIT+C-BEGIN request was issued on any dialogue:

    - invoke the "Rollback next transaction" procedure (see 11.5.17);

  ii) a *commit request* was sent on this dialogue and no *commit-confirm* has been received:

    - issue a CAF-PLEASE reques with the AE-Title of the CAF-PLEASE request set to the value of the AE-Title taken from the branch identifier for this neighbour, contained in the log-commit record;

  iii) always:

    - continue.

d) DECIDED (rollback) state:

  i) if this is the superior dialogue and:

    1) if a TP-U-ABORT request was received and the *last rollback confirm was received*:

      - invoke the "Initiating transaction after rollback" procedure (see 11.5.11);

    2) otherwise:

      - continue;

  ii) if this is a subordinate dialogue and neither a *rollback confirm* was received nor a *rollback response* was issued, the *last rollback confirm was received*, and:

    1) if this is an intermediate node and the superior *dialogue has not been detached*:

      - invoke the "Reporting rollback to superior" procedure (see 11.5.16);

    2) if the *superior dialogue is not chaining* and *rollback reporting has completed*:

      - invoke the "Initiating transaction after rollback" procedure (see 11.5.11);

  iii) otherwise:

    - continue.

e) ONE-PHASE state and a C-NOCHANGE request or AF-NOCHANGE request was issued on the dialogue:

  - enter the DECIDED (unknown) state;
  - issue a TP-UNKNOWN indication;
  - invoke the "Sending not-determined result from a ONE-PHASE or READ-ONLY node" procedure (see 11.5.19) if a C-NOCHANGE indication has been received and no C-NOCHANGE response has been issued on any subordinate dialogue which is still *attached*.

f) READ-ONLY state or EARLY-EXIT state and this is the superior dialogue:

  - enter the DECIDED (unknown) state;
  - issue a TP-UNKNOWN indication.

If this is a superior dialogue and a *one-phase indication has been received* on the dialogue and:

a) The TPPM is in the DECIDED (commit) state and the Chained functional unit is selected on the dialogue:

  – invoke the "Rollback next transaction" procedure (see 11.5.17).

b) The TPPM is in the DECIDED (rollback) state, a TP-U-ABORT request was received and the *last rollback confirm was received*:

  – invoke the "Initiating transaction after rollback" procedure (see 11.5.11).

c) The TPPM is in the DECIDED (commit-one-phase) or DECIDED (unknown) state and a TP-U-ABORT request was received:

  – invoke the "Completing ONE-PHASE and READ-ONLY" procedure (see 11.5.5).

11.3.22 Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm on a channel

If this is a protocol error or internal error and the association has not been aborted:

– issue an AF-ABORT (provider, abortRI) request with the Diagnostic parameter set to:

  1) "protocol error", if this is a protocol error;

  2) "transient-failure" or "permanent-failure", based on a local decision, if this is an internal error.

If the channel was with the neighbour to which a *ready signal was sent* and the TPPM is in the:

a) READY state:

  – issue a CAF-PLEASE request with the AE-Title of the CAF-PLEASE request set to the value of the AE-Title taken from the branch identifier for the neighbour, contained in the log-ready record.

b) DECIDED (commit) state:

  – continue.

If the channel was with a neighbour from whom a *ready signal was received* and the TPPM is in the:

a) READY state:

  – continue.

b) DECIDED (commit) state:

  – issue a CAF-PLEASE request with the AE-Title of the CAF-PLEASE request set to the value of the AE-Title taken from the branch identifier for the neighbour, contained in the log-ready record.

11.3.23 Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm (CPM)

If this is a protocol error or internal error:

– issue an AF-ABORT (provider, abortRI) request with the Diagnostic parameter set to:

  1) "protocol error", if this is a protocol error;

  2) "transient-failure" or "permanent-failure", based on a local decision, if this is an internal error.

If a *CAF-PLEASE request is outstanding* for the channel:

– invoke the "Fail an outstanding CAF-PLEASE request" procedure (see 11.5.7).

11.3.24 TP-GRANT-CONTROL request

– Invoke the "First request/response" procedure (see 11.5.8), if this is a superior dialogue and if no AF-BEGIN-DIALOGUE response has been issued on the dialogue since an AF-BEGIN-DIALOGUE indication has been received;

– issue an AF-GRANT-CONTROL request.

11.3.25 AF-GRANT-CONTROL indication

If the dialogue is not coordinated:

– issue a TP-GRANT-CONTROL indication.

If the dialogue is coordinated and the TPPM is in the:

a) ACTIVE state:

  – issue a TP-GRANT-CONTROL indication.

b) DECIDED (rollback) state:

  – continue.

11.3.26 TP-REQUEST-CONTROL request

– Invoke the "First request/response" procedure (see 11.5.8), if this is a superior dialogue and if no AF-BEGIN-DIALOGUE response has been issued on the dialogue since an AF-BEGIN-DIALOGUE indication has been received;

– issue an AF-REQUEST-CONTROL request.

11.3.27 AF-REQUEST-CONTROL indication

– Continue if at least one of the following conditions is met:

  a) the TPPM does not have the control of the dialogue;

  b) the TPPM has an outstanding dialogue termination request;

  c) the TPPM has already invoked an AF-PREPARE request for the current provider-supported transaction;

  d) a *transaction completion request* has been received;

  e) the TPPM is in the DECIDED (rollback) state.

Otherwise:

– issue a TP-REQUEST-CONTROL indication.

11.3.28 TP-HANDSHAKE request

– Invoke the "First request/response" procedure (see 11.5.8), if this is a superior dialogue and if no AF-BEGIN-DIALOGUE response has been issued on the dialogue since an AF-BEGIN-DIALOGUE indication has been received;

– issue an AF-HANDSHAKE request.

11.3.29 AF-HANDSHAKE indication

Do only the first applicable action of the following:

a) If the TPPM is in the DECIDED (rollback) state and the dialogue is coordinated:

  – continue.

b) If the Shared Control functional unit is selected and the TPPM is in a *user error purging period*:

  – continue.

c) If the TPPM is in the ACTIVE state, a C-READY indication was not received, the Shared Control functional unit is selected and either a *transaction completion request* has been received or a TP-PREPARE request has been received for this dialogue:

  – invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "user-data-transaction-completion-collision".

d) READY, ONE-PHASE or READ-ONLY state and a C-READY request *or ready-substitute request* was sent on this dialogue, no AF-PREPARE indication was received, the Shared Control Functional Unit is selected and at least one of the Dynamic Commit, Implicit Prepare or One-phase Commit functional units were selected:

  – continue.

  NOTE – This can only occur if either this or the peer TPSUI has issued primitives in violation of the application semantics. The peer TPPM will detect the error when it receives the C-READY indication (or *ready-substitute indication*).

e) If the TPPM is not in a *user error purging period*:

  - issue a TP-HANDSHAKE indication.

f) If the Polarized Control functional unit was selected and the TPPM is in a *user error purging period*:

  - issue a TP-HANDSHAKE indication.

11.3.30 TP-HANDSHAKE response

- Invoke the "First request/response" procedure (see 11.5.8), if this is a superior dialogue and if no AF-BEGIN-DIALOGUE response has been issued on the dialogue since an AF-BEGIN-DIALOGUE indication has been received;
- issue an AF-HANDSHAKE response.

11.3.31 AF-HANDSHAKE confirm

If the dialogue is not coordinated:

- issue a TP-HANDSHAKE confirm.

If the dialogue is coordinated and the TPPM is in the:

a) ACTIVE state:

  - issue a TP-HANDSHAKE confirm.

b) DECIDED (rollback) state:

  - continue.

c) READY, ONE-PHASE or READ-ONLY states if a C-READY request *or ready-substitute request* was issued on this dialogue and no AF-PREPARE indication was received and the Shared Control Functional Unit is selected:

  - invoke the "User protocol error" procedure (see 11.5.21).

  NOTE – This can only occur if either this or the peer TPSUI has issued primitives in violation of the application semantics, when one or more of Dynamic Commit, Implicit Prepare or One-phase Commit functional units have been selected. In these cases it can be impossible for the peer TPPM to detect the error, and thus the TPSP cannot guarantee the propagation of a rollback. An AF-HANDSHAKE confirm received other than under these conditions implies the peer TPPM is at fault, and is treated as a protocol error.

11.3.32 TP-HANDSHAKE-AND-GRANT-CONTROL request

- Invoke the "First request/response" procedure (see 11.5.8), if this is a superior dialogue and if no AF-BEGIN-DIALOGUE response has been issued on the dialogue since an AF-BEGIN-DIALOGUE indication has been received;
- issue an AF-HANDSHAKE-AND-GRANT-CONTROL request.

11.3.33 AF-HANDSHAKE-AND-GRANT-CONTROL indication

If the dialogue is not coordinated:

- issue a TP-HANDSHAKE-AND-GRANT-CONTROL indication.

If the dialogue is coordinated and the TPPM is in the:

a) ACTIVE state:

  - issue a TP-HANDSHAKE-AND-GRANT-CONTROL indication

b) DECIDED (rollback) state:

  - continue.

c) READY, ONE-PHASE or READ-ONLY states if a C-READY request *or ready-substitute request* was issued on this dialogue and no AF-PREPARE indication was received and the Shared Control Functional Unit is selected:

  - invoke the "User protocol error" procedure (see 11.5.21).

NOTE – This can only occur if either this or the peer TPSUI has issued primitives in violation of the application semantics, when one or more of Dynamic Commit, Implicit Prepare or One-phase Commit functional units have been selected. In these cases it can be impossible for the peer TPPM to detect the error, and thus the TPSP cannot guarantee the propagation of a rollback. An AF-HANDSHAKE-AND-GRANT-CONTROL indication received other than under these conditions implies the peer TPPM is at fault, and is treated as a protocol error.

11.3.34 TP-HANDSHAKE-AND-GRANT-CONTROL response

– Invoke the "First request/response" procedure (see 11.5.8), if this is a superior dialogue and if no AF-BEGIN-DIALOGUE response has been issued on the dialogue since an AF-BEGIN-DIALOGUE indication has been received;

– issue an AF-HANDSHAKE-AND-GRANT-CONTROL response.

11.3.35 AF-HANDSHAKE-AND-GRANT-CONTROL confirm

If the dialogue is not coordinated:

– issue a TP-HANDSHAKE-AND-GRANT-CONTROL confirm.

If the dialogue is coordinated and the TPPM is in the:

a) ACTIVE state:

  – issue a TP-HANDSHAKE-AND-GRANT-CONTROL confirm.

b) DECIDED (rollback) state:

  – continue.

11.3.36 TP-BEGIN-TRANSACTION request

NOTE – The Session Layer synchronize-minor token is needed in order to issue this request. The TPPM guarantees that this *token* will be positioned with the issuer unless the TPSUI or U-ASE have moved it, in which case it is the responsibility of the TPSUI to get the *token* before issuing this request. This is a constraint on the movement of the *token* by the U-ASE, described in Annex B.

– Invoke the "Initiating a transaction branch" procedure (see 11.5.9).

11.3.37 C-BEGIN indication or AF-BEGIN-TRANSACTION indication

The indication shall be from the superior in the dialogue tree. At least one of the Commit and One-phase commit functional units shall be selected. At least one of the following conditions shall be true:

a) the coordination level is "none";

b) the TPPM is in the DECIDED (rollback), DECIDED (commit-one-phase) or DECIDED (unknown) states and the *dialogue is chaining;* or

c) the One-phase commit functional unit is selected, the Commit functional unit is not selected and the *dialogue is chaining.*

If the Shared Control functional unit is selected and a dialogue termination request is outstanding:

– issue a TP-P-ABORT indication with the Diagnostic parameter set to "begin-transaction-end-dialogue-collision" and the Rollback parameter set to "false";

– issue a C-ROLLBACK request;

– issue an SAF-DETACH-ASSOCIATION (rollback-confirm-expected) request.

If the TPPM is a root in a transaction tree and a dialogue termination request is not outstanding:

– issue an AF-ABORT (provider, rollbackRI) request with the Diagnostic parameter set to "begin-transaction-reject";

– issue a TP-P-ABORT indication with the value of the Rollback parameter set to "false"; and with the Diagnostic parameter set to "begin-transaction-reject";

– issue an SAF-DETACH-ASSOCIATION (rollback-confirm-expected) request.

NOTE 1 – An implementation may choose to wait for the current transaction to complete and then accept the C-BEGIN indication. From a conformance testing view, this is equivalent to not being able to process the C-BEGIN indication until the current transaction has completed (when it can then be successfully processed, instead of rejected). The TPPM procedures and state tables do not provide procedures to allow waiting for the transaction to complete.

If the TPPM is not a root in the transaction tree and a dialogue termination request is not outstanding:

a) If a TP-BEGIN-DIALOGUE indication has been issued and the TPPM is not in a transaction:

   – issue a TP-BEGIN-TRANSACTION indication;

   – become a leaf node;

   – enter the ACTIVE state.

b) If a TP-BEGIN-DIALOGUE indication has not been issued:

   – create a TPSUI of the type specified by the Recipient-TPSU-Title parameter of the AF-BEGIN-DIALOGUE indication, or a TPSUI of a default type if the parameter is not present.

   NOTE 2 – From the OSIE perspective, whether a new TPSUI is "created" in a real open system or whether an old instance is re-used, is a local matter.

   – issue a TP-BEGIN-DIALOGUE indication with the parameters as specified in the previously received AF-BEGIN-DIALOGUE indication;

   NOTE 3 – If the *dialogue will be coordinated,* the TP-BEGIN-DIALOGUE indication is issued as above since the C-BEGIN indication has arrived.

   – become a leaf node;

   – enter the ACTIVE state.

c) If an AF-BEGIN-DIALOGUE response was issued:

   – issue a C-BEGIN response.

d) If the TPPM is in either the READ-ONLY or EARLY-EXIT state and no TP-UNKNOWN indication or TP-COMMIT indication has been issued:

   – enter the DECIDED (unknown) state;

   – issue a TP-UNKNOWN indication.

   NOTE 4 – This causes a *TP-DONE request to be owed*, thus action e) below applies.

e) If the TPPM is in any of the DECIDED (rollback), DECIDED (commit-one-phase) or DECIDED (unknown) states and *a TP-DONE request is owed*:

   – *close the PSAP*.

f) If the TPPM is in the DECIDED (rollback) state, and *a TP-DONE request is not owed*:

   – invoke the "Initiating transaction after rollback" procedure (see 11.5.11).

g) If the TPPM is in the DECIDED (commit-one-phase) state or DECIDED (unknown) state and a *TP-DONE request is not owed*:

   – invoke the "Completing ONE-PHASE and READ-ONLY" procedure (see 11.5.5).

h) If the TPPM is in the DECIDED (commit) state:

   – invoke the "Completing commitment" procedure (see 11.5.3).

11.3.38 C-BEGIN confirm

– Continue.

11.3.39 TP-DATA request

– Invoke the "First request/response" procedure (see 11.5.8), if this is a superior dialogue and if no AF-BEGIN-DIALOGUE response has been issued on the dialogue since an AF-BEGIN-DIALOGUE indication has been received;

– issue a U-ASE request.

11.3.40 U-ASE indication

If the TPPM is in a *user error purging period*:

– continue.

Otherwise:

a) If the dialogue is not coordinated:

   – issue a TP-DATA indication.

b) If the dialogue is coordinated and the TPPM is in the:

   1) ACTIVE state:

      i) and a *transaction completion request* has been received, and this indication occurred on a dialogue with the Shared Control functional unit selected or on a dialogue with the Polarized Control functional unit selected after an AF-PREPARE request with the Data-Permitted parameter set to "true" has been issued:

         – invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "user-data-transaction-completion-collision",

      ii) otherwise:

         – issue a TP-DATA indication.

   2) DECIDED (rollback) state:

      – continue.

   3) READY, ONE-PHASE or READ-ONLY states if a C-READY request *or ready-substitute request* was issued on this dialogue and no AF-PREPARE indication was received and the Shared Control Functional Unit is selected:

      – invoke the "User protocol error" procedure (see 11.5.21).

      NOTE – This can only occur if either this or the peer TPSUI has issued primitives in violation of the application semantics, when one or more of Dynamic Commit, Implicit Prepare or One-phase Commit functional units have been selected. In these cases it can be impossible for the peer TPPM to detect the error, and thus the TPSP cannot guarantee the propagation of a rollback. A U-ASE indication received other than under these conditions implies the peer TPPM is at fault, and is treated as a protocol error.

11.3.41 TP-DEFERRED-END-DIALOGUE request

If the Implicit Prepare functional unit is selected, or optionally if it is not selected, then according to a local decision:

– issue an AF-DEFER (end-dialogue) request.

NOTE – It is an implementation option, except when the Implicit Prepare functional unit is selected, to either invoke the AF-DEFER request immediately or to delay it until the AF-PREPARE request is invoked.

11.3.42 TP-DEFERRED-GRANT-CONTROL request

If the Implicit Prepare functional unit is selected, or optionally if it is not selected, then according to a local decision:

– issue an AF-DEFER (grant-control) request.

NOTE – It is an implementation option, except when the Implicit Prepare functional unit is selected, to either invoke the AF-DEFER request immediately or to delay it until the AF-PREPARE request is invoked.

11.3.43 AF-DEFER indication

If the dialogue is coordinated, and the TPPM is in the:

a) ACTIVE state and no *transaction completion request* has been received, either:

   – issue a TP-DEFERRED-END-DIALOGUE indication if the Type parameter is "end-dialogue"; or

   – issue a TP-DEFERRED-GRANT-CONTROL indication if the Type parameter is "grant-control".

b) ACTIVE state and a *transaction completion request* has been received:

   – invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "user-data-transaction-completion-collision".

c) DECIDED (rollback) state:

   – continue.

d) READY, ONE-PHASE or READ-ONLY states if a C-READY request or *ready-substitute request* was issued on this dialogue and no AF-PREPARE indication was received and the Shared Control Functional Unit is selected:

   – invoke the "User protocol error" procedure (see 11.5.21).

   NOTE – This can only occur if either this or the peer TPSUI has issued primitives in violation of the application semantics, when one or more of Dynamic Commit, Implicit Prepare or One-phase Commit functional units have been selected. In these cases it can be impossible for the peer TPPM to detect the error, and thus the TPSP cannot guarantee the propagation of a rollback. An AF-DEFER indication received other than under these conditions, implies the peer TPPM is at fault, and is treated as a protocol error.

11.3.44 TP-PREPARE request

If a TP-DEFERRED-END-DIALOGUE request was received and no AF-DEFER (end-dialogue) request was issued:

– issue an AF-DEFER (end-dialogue) request.

If no TP-DEFERRED-END-DIALOGUE request was received, a TP-DEFERRED-GRANT-CONTROL request was received and no AF-DEFER (grant-control) request was issued:

– issue an AF-DEFER (grant-control) request.

Always:

– issue an AF-PREPARE request.

11.3.45 TP-COMMIT request

Optionally:

– commence setting the TPPM bound data to the ready-to-commit-state.

For each subordinate to which an AF-PREPARE request has not already been issued:

a) If a TP-DEFERRED-END-DIALOGUE request was received and no AF-DEFER (end-dialogue) request was issued:

   – issue an AF-DEFER (end-dialogue) request.

b) If no TP-DEFERRED-END-DIALOGUE request was received, a TP-DEFERRED-GRANT-CONTROL request was received and no AF-DEFER (grant-control) request was issued:

   – issue an AF-DEFER (grant-control) request.

If the *last ready was received*:

– invoke the "Making commitment decision" procedure (see 11.5.12).

If the *ready state may be entered*:

– invoke the "Entering READY state" procedure (see 11.5.6).

If the *last ready has not been received* and *the ready state may not be entered*, for each dialogue supporting a branch of the transaction on which no AF-PREPARE request has been issued and no C-READY indication or *ready-substitute indication* has been received, and at least one of the following is true:

i) *ready is receivable* and the dialogue is to a subordinate and the Implicit Prepare functional unit is not selected; or

ii) ready is receivable; and the Polarized Control functional unit is selected and an AF-PREPARE (no-data-permitted) indication has not been received; or

iii) a local decision determines.

   NOTE – It is expected this local decision will be configurable for each dialogue. The effect of iii) being true is equivalent to combining, as a single action, the *completion request* with TP-PREPARE requests for the dialogues for which i) and ii) are false.

Then:

- issue an AF-PREPARE request with the Data-Permitted parameter:
  1) absent, if the Shared Control functional unit is selected; or
  2) set to "false", if the Polarized Control functional unit is selected.

11.3.46 AF-PREPARE indication

If the coordination level is "commitment" and TPPM is in the:

a) ACTIVE state, and:

   1) the TPPM is in a *user error purging period* or a handshake request is outstanding:

      NOTE 1 – A handshake request outstanding may occur only if the Shared Control functional unit was selected.

      - invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "user-data-transaction-completion-collision",

   2) otherwise:

      i) if *transaction completion request* has not been received:

         - issue a TP-PREPARE indication,

      ii) if the *READY state may be entered*:

         - invoke the "Entering READY state" procedure (see 11.5.6),

      iii) if the *Non-recovery states may be entered*:

         - invoke the " Entering ONE-PHASE or READ-ONLY state" procedure (see 11.5.20),

      iv) otherwise:

         - continue.

b) DECIDED (rollback), READY, ONE-PHASE or READ-ONLY state:

   - continue.

   NOTE 2 – This will only occur for the READY, ONE-PHASE and READ-ONLY states if the neighbour TPSUI issued a TP-PREPARE request when this TPSUI was already permitted to issue a *transaction completion request* according to the application semantics. This must either be a subordinate dialogue with the Dynamic Commit functional unit selected or the Implicit Prepare functional unit is selected.

11.3.47 C-READY indication

If the TPPM is in the ACTIVE state, perform the first applicable of the following sets of actions:

a) If the TPPM is in a *user error purging period* or a handshake request is outstanding:

   NOTE 1 – A handshake request outstanding may occur only if the Shared Control functional unit was selected; a C-READY indication can only be received under either of these conditions if at least one of the Dynamic Commit and Implicit Prepare functional units were selected.

   - invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "user-data-transaction-completion-collision".

b) If no *completion request* was received:

   - issue a TP-READY indication.

c) If the *last ready was received*:

   - invoke the "Making commitment decision" procedure (see 11.5.12).

d) If the *ready state may be entered*:

   - invoke the "Entering READY state" procedure (see 11.5.6).

e) Otherwise:

   - continue.

If the TPPM is in the READY state and *the token is owned*:

– invoke the "Making commitment decision" procedure (see 11.5.12).

If the TPPM is in the READY state and *the token is not owned*:

– continue.

If the TPPM is in the ONE-PHASE or READ-ONLY states:

– continue.

NOTE 2 – This only occurs following a READY/ONE-PHASE or READY/READ-ONLY collision.

If the TPPM is in the DECIDED (rollback) state:

– continue.

NOTE 3 – This only occurs on a superior dialogue.

11.3.48 C-COMMIT indication or C-COMMIT+C-BEGIN indication

The TPPM shall be in the READY state:

– invoke the "Receiving commit order" procedure (see 11.5.14).

If a C-READY indication or *ready-substitute indication* has been received on any other dialogue that is still in the transaction:

– invoke the "Sending commit order" procedure (see 11.5.18).

11.3.49 AF-ABORT (user, commitRI) indication

The TPPM shall be in the READY state:

– invoke the "Receiving commit order" procedure (see 11.5.14);

– issue a TP-U-ABORT indication with the value of the Rollback parameter set to "false" if no TP-U-ABORT request was received.

If a C-READY indication or C-NOCHANGE indication has been received on any other dialogue:

– invoke the "Sending commit order" procedure (see 11.5.18).

11.3.50 TP-DONE request

If the Heuristic-Report parameter is specified:

– invoke the "Recording the heuristic condition" procedure (see 11.5.15).

If the TPPM is in the DECIDED (commit) state and:

a) If a *ready signal was received* or a *one-phase indication* was received from the superior, the *reporting status is known, reporting applies* on the branch to the superior and *reports have not been sent*:

   – invoke the "Reporting on the commit-coordinator:root path" procedure (see 11.5.4).

b) The *last commit confirm was received* and the *reporting status is known*:

   – invoke the "Confirm and completing commitment" procedure (see 11.5.1).

c) If none of the above conditions apply:

   – continue.

If the TPPM is in the DECIDED (commit-one-phase) state:

a) If the *reporting status is known* and *reporting applies* on the branch to the superior and *reports have not been sent*:

   – invoke the "Reporting on the commit-coordinator:root path" procedure (see 11.5.4).

b) If the *reporting status is known*:

   – invoke the "Completing ONE-PHASE and READ-ONLY" procedure (see 11.5.5).

c) If none of the above conditions apply:

   – continue.

If the TPPM is in the DECIDED (unknown) and either the *superior dialogue is not chaining* or a C-BEGIN indication has been received:

– invoke the "Completing ONE-PHASE and READ-ONLY" procedure (see 11.5.5).

If the TPPM is in the DECIDED (rollback) state and:

a) The *last rollback confirm was received*:

   1) if the TPPM is an intermediate or a leaf and the *superior dialogue has not been detached* and the TPPM has not issued a *rollback request* or a *rollback response* to the superior:

      – invoke the "Reporting rollback to superior" procedure (see 11.5.16);

   2) if the *superior dialogue is not chaining* and *rollback reporting has completed*:

      – invoke the "Initiating transaction after rollback" procedure (see 11.5.11);

   3) if the TPPM is an intermediate and a C-BEGIN indication has been received and the *dialogue has not been detached*:

      – invoke the "Initiating transaction after rollback" procedure (see 11.5.11);

   4) otherwise:

      – continue.

b) The *last rollback confirm was not received*:

   – continue.

11.3.51 C-COMMIT confirm or AF-REPORT (commitRC) indication

If this is an AF-REPORT indication, the Heuristic Containment Required functional unit is not selected on the dialogue and the Heuristic Report parameter does not have the value "none":

– issue a TP-HEURISTIC-REPORT indication;

– invoke the "Recording the heuristic condition" procedure (see 11.5.15).

If this is an AF-REPORT indication and the Completion diagnostics functional unit is selected on this dialogue and any of the Severity, Diagnostic or Completion-data parameters are present:

– issue a TP-COMPLETION-REPORT indication.

If this is a C-COMMIT confirm on the superior dialogue and:

1) If a TP-U-ABORT request was received:

   – issue an AF-ABORT (user, dataRI) request;

   – issue an SAF-DETACH-ASSOCIATION (begin-fear) request.

2) If no TP-U-ABORT request was received and either an AF-DEFER (end-dialogue) indication was received or, an AF-ABORT (user,commitRI) request was issued:

   – issue an SAF-DETACH-ASSOCIATION (free) request.

3) If neither of the above conditions apply:

   – continue.

If this is a subordinate dialogue and either:

a) a TP-DEFERRED-END-DIALOGUE request was received and no TP-U-ABORT request was received; or

b) an AF-ABORT (user, commitRI) request was issued,

then:

– issue an SAF-DETACH-ASSOCIATION (free) request.

If this is a subordinate dialogue and a TP-U-ABORT request was received:

a) If a C-COMMIT+C-BEGIN request was issued:

   – issue an AF-ABORT (user, rollbackRI) request;

   – issue an SAF-DETACH-ASSOCIATION (rollback-confirm-expected) request.

b) If no C-COMMIT+C-BEGIN request was issued:

   – issue an AF-ABORT (user, dataRI) request, if no AF-ABORT (user, commitRI) request and no AF-DEFERRED-END-DIALOGUE request was issued;

   – issue an SAF-DETACH-ASSOCIATION (free) request.

If this is a subordinate dialogue and no TP-U-ABORT request was received and no TP-DEFERRED-END-DIALOGUE request was received:

– issue a C-ROLLBACK request, if a C-COMMIT+C-BEGIN request was issued and a TP-ROLLBACK indication is pending;

– *close the PSAP*.

NOTE 1 – Any further events to/from the subordinate are to be handled either as part of the next transaction branch, or when this transaction branch is completed and the dialogue returns to coordination level of "none". In this way, the events can be handled as part of the normal procedures (e.g. the TPPM is in the ACTIVE state), rather than within the scope of the transaction termination procedures.

If a *ready signal was received* or a *one-phase indication* was received from the superior, the *reporting status is known, reporting applies* on the branch to the superior and *reports have not been sent*:

– invoke the "Reporting on the commit-coordinator:root path" procedure (see 11.5.4).

NOTE 2 – The superior will not confirm commitment until it receives the report.

If the *last commit confirm was received*:

– invoke the "Confirm and complete commitment" procedure (see 11.5.1).

11.3.52 AF-ABORT (user, commitRC) indication or AF-ABORT-AND-REPORT (commitRC) indication

A C-COMMIT request was issued and the Unchained Transactions functional unit was selected, or a C-COMMIT+ C-BEGIN request was issued and the Chained Transactions functional unit was selected.

Always:

– issue an SAF-DETACH-ASSOCIATION (free) request if a C-COMMIT request was issued;

– issue an SAF-DETACH-ASSOCIATION (rollback-indication-expected) request if a C-COMMIT+ C-BEGIN request was issued.

If no TP-U-ABORT request was received:

– issue a TP-U-ABORT indication with the value of the Rollback parameter set to "false".

If this is an AF-ABORT-AND-REPORT indication, the Heuristic Containment Required functional unit is not selected on the dialogue and the Heuristic Report parameter does not have the value "none":

– issue a TP-HEURISTIC-REPORT indication;

– invoke the "Recording the heuristic condition" procedure (see 11.5.15).

If this is an AF-ABORT-AND-REPORT indication and the Completion diagnostics functional unit is selected on this dialogue and any of the Severity, Diagnostic or Completion-data parameters are present:

– issue a TP-COMPLETION-REPORT indication with the Completion-data parameter set to the value of the Completion-data parameter of the AF-ABORT-AND-REPORT indication.

If the *last commit confirm* was received and the *reporting status is known*:

– invoke the "Confirm and complete commitment" procedure (see 11.5.1).

11.3.53 TP-ROLLBACK request

– Invoke the "Initiating rollback at TPPM" procedure (see 11.5.10).

11.3.54 C-ROLLBACK indication or AF-REPORT (rollbackRI) indication

If the TPPM is in the:

a) ACTIVE, READY, ONE-PHASE, READ-ONLY or EARLY-EXIT state:

   – invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "superior-rollback" if this is a a superior dialogue and a diagnostic-value of "subordinate-rollback" if this is a subordinate dialogue.

b) ACTIVE state:

   – *close the PSAP*, if this is a subordinate dialogue and the Unchained Transactions functional unit is selected on this dialogue.

c) Any state, and this is an AF-REPORT indication, the Heuristic Containment Required functional unit is not selected on the dialogue and the Heuristic Report parameter does not have the value "none":

   – issue a TP-HEURISTIC-REPORT indication;

   – invoke the "Recording the heuristic condition" procedure (see 11.5.15).

d) Any state, and this is an AF-REPORT indication and the Completion Diagnostics functional unit is selected on this (subordinate) dialogue and any of the Severity, Diagnostic or Completion-data parameters are present and non-empty:

   – issue a TP-COMPLETION-REPORT indication.

e) DECIDED (rollback) state:

   1) if the dialogue is with the subordinate and a TP-U-ABORT request was received:

      – issue an AF-ABORT (user, rollbackRC) request;

      – issue an SAF-DETACH-ASSOCIATION (free) request;

   2) if the dialogue is with the subordinate and a TP-U-ABORT request was not received:

      – issue a C-ROLLBACK response;

      – *close the PSAP*, if the Unchained Transactions functional unit is selected on this dialogue;

   3) if either of the following are true:

      i) the dialogue is with the superior and a *rollback request* has been issued to the superior; or

      ii) the dialogue is with a subordinate, the *last rollback confirm was received*, this is an intermediate node, and the superior *dialogue has not been detached*,

      then:

      – invoke the "Reporting rollback to superior" procedure (see 11.5.16);

   4) if the *last rollback confirm was received* and the *superior dialogue is not chaining* and *rollback reporting has completed*:

      – invoke the "Initiating transaction after rollback" procedure (see 11.5.11);

   5) if the dialogue is with the superior and the *last rollback confirm was not received*:

      – continue.

11.3.55 C-CANCEL indication

If the TPPM is in the:

a) ACTIVE, READY, ONE-PHASE or READ-ONLY state:

   – invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "subordinate-rollback" if this is subordinate dialogue or "superior-rollback" if this is the superior dialogue.

b) DECIDED (rollback) state:

   – continue.

11.3.56 AF-ABORT (user/provider, rollbackRI) indication or AF-ABORT-AND-REPORT (rollbackRI) indication

If this is an AF-ABORT (provider, rollbackRI) indication, then the Diagnostic parameter shall be set to "begin-transaction-reject", this shall be a subordinate dialogue, the Unchained Transactions functional unit shall have been selected, and a C-BEGIN confirm shall not have been received.

If the dialogue is coordinated and the TPPM is in the:

a) ACTIVE state and has a *dialogue establishment indication outstanding* and an AF-ABORT (user/provider, rollbackRI) indication is received:

   – issue a TP-U-ABORT indication with the Rollback parameter set to "false" if the Type parameter is "user";

   – issue a TP-P-ABORT indication with the Rollback parameter set to "false" if the Type parameter is "provider";

   – issue a C-ROLLBACK response;

- issue an SAF-DETACH-ASSOCIATION (free) request.

  NOTE – The procedure "Initiating rollback at TPPM" (see 11.5.10) is not triggered in this case because the transaction cannot have started at the subordinate"s node.

b) ACTIVE or READY state and does not have a *dialogue establishment indication outstanding*:

   - issue a TP-U-ABORT indication with the Rollback parameter set to "true" if the Type parameter is "user" or this is an AF-ABORT-AND-REPORT indication, and no TP-U-ABORT request was received;
   - issue a TP-P-ABORT indication with the Rollback parameter set to "true" if the Type parameter is "provider" and no TP-U-ABORT request was received;
   - invoke the "Initiating rollback at TPPM" procedure (see 11.5.10);
   - issue an SAF-DETACH-ASSOCIATION (free) request if an AF-ABORT (user/provider, rollbackRI) indication was received from a subordinate.

c) DECIDED (rollback) state:

   1) if a TP-U-ABORT request was received:
      - continue;
   2) if no TP-U-ABORT request was received:
      - issue a TP-P-ABORT indication with the Rollback parameter set to "false" if the type is "provider";
      - issue a TP-U-ABORT indication with the Rollback parameter set to "false" if the type is "user" or this is an AF-ABORT-AND-REPORT indication;
   3) if the dialogue is with a subordinate:
      - issue a C-ROLLBACK response;
      - issue an SAF-DETACH-ASSOCIATION (free) request.

d) Any state, and this is an AF-ABORT-AND-REPORT indication, the Heuristic Containment Required functional unit is not selected on the dialogue and the Heuristic Report parameter does not have the value "none":

   - issue a TP-HEURISTIC-REPORT indication;
   - invoke the "Recording the heuristic condition" procedure (see 11.5.15).

e) Any state, and this is an AF-ABORT-AND-REPORT indication and the Completion Diagnostics functional unit is selected on this (subordinate) dialogue and any of the Severity, Diagnostic or Completion-data parameters are present:

   - issue a TP-COMPLETION-REPORT indication, with the Completion-data parameter set to the value of the Completion-data parameter of the AF-ABORT-AND-REPORT indication.

f) DECIDED (rollback) state:

   1) if either of the following is true:

      i) the dialogue is with the superior and a *rollback request* has been issued to the superior; or

      ii) the dialogue is with a subordinate, the *last rollback confirm was received*, this is an intermediate node, and the superior *dialogue has not been detached*,

      then:

      - invoke the "Reporting rollback to superior" procedure (see 11.5.16);

   2) if the *last rollback confirm was received* and the *superior dialogue is not chaining* and *rollback reporting has completed*:

      - invoke the "Initiating transaction after rollback" procedure (see 11.5.11).

11.3.57 C-ROLLBACK confirm or AF-REPORT (rollbackRC) indication

The TPPM shall be in the DECIDED (rollback) state or in the EARLY-EXIT state.

If this was an AF-REPORT indication, the Heuristic Containment Required functional unit is not selected on the dialogue and the Heuristic Report parameter does not have the value "none":

- issue a TP-HEURISTIC-REPORT indication;
- invoke the "Recording the heuristic condition" procedure (see 11.5.15).

If this was an AF-REPORT indication and the Completion Diagnostics functional unit is selected on this (subordinate) dialogue and any of the Severity, Diagnostic or Completion-data parameters are present:

– issue a TP-COMPLETION-REPORT indication.

If the *rollback confirm* was from the subordinate:

a) If a TP-U-ABORT request has been received and no AF-ABORT request was issued:

   – issue an AF-ABORT (user, dataRI) request.

b) If the Unchained Transactions functional unit is selected on this dialogue and no TP-U-ABORT request was received:

   – *close the PSAP*.

c) If the *last rollback confirm was received* and this is an intermediate node and the superior *dialogue has not been detached*:

   – invoke the "Reporting rollback to superior" procedure (see 11.5.16).

d) If the *last rollback confirm was received* and the *superior dialogue is not chaining* and *rollback reporting has completed*:

   – invoke the "Initiating transaction after rollback" procedure (see 11.5.11).

e) If a TP-U-ABORT request was received or a TP-U-ABORT indication was issued:

   – issue an SAF-DETACH-ASSOCIATION (free) request;

f) If none of the above conditions is satisfied:

   – continue.

If the *rollback confirm* was from the superior and the TPPM is in the DECIDED (rollback) state:

a) If the superior *dialogue is not chaining* and *a TP-DONE request is not owed*:

   – invoke the "Initiating transaction after rollback" procedure (see 11.5.11).

b) If a TP-U-ABORT request was received:

   – issue an AF-ABORT (user, dataRI) request if an AF-ABORT (user, rollbackRI) request has not been issued;

   – issue an SAF-DETACH-ASSOCIATION (begin-fear) request if no AF-ABORT (user, rollbackRI) request was issued;

   – issue an SAF-DETACH-ASSOCIATION (free) request if an AF-ABORT (user, rollbackRI) request was issued.

c) If the superior *dialogue has not been detached*, the Unchained Transactions functional unit is selected, and *a TP-DONE request is owed*:

   – *close the PSAP*.

d) If none of the above conditions is satisfied:

   – continue.

If the TPPM is in the EARLY-EXIT state:

a) Always:

   – invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "superior-rollback".

b) If a TP-U-ABORT request was received:

   – issue an AF-ABORT (user, dataRI) request;

   – issue an SAF-DETACH-ASSOCIATION (begin-fear) request.

c) If no TP-U-ABORT request was received:

   – *close the PSAP*.

11.3.58 AF-ABORT (user/provider, rollbackRC) indication or AF-ABORT-AND-REPORT (rollbackRC) indication

The TPPM shall be in the DECIDED (rollback) state and no AF-ABORT (user, rollbackRI) request shall have been issued on this dialogue or in the EARLY-EXIT state.

If the dialogue is coordinated and the TPPM is in the DECIDED (rollback) state and one of the following is true:

a) this is an AF-ABORT indication on a subordinate dialogue for which the Unchained Transactions functional unit has been selected, the Diagnostic parameter is "Begin-transaction-reject", the Type parameter is set to "provider", and a C-BEGIN confirm has not been received; or

b) this is an AF-ABORT indication and the Type parameter is set to "user"; or

c) this is an AF-ABORT-AND-REPORT indication.

Then:

a) Always:

   – issue an SAF-DETACH-ASSOCIATION (free) request.

b) If this is an AF-ABORT-AND-REPORT indication, the Heuristic Containment Required functional unit is not selected on the dialogue and the Heuristic Report parameter does not have the value "none":

   – issue a TP-HEURISTIC-REPORT indication;

   – invoke the "Recording the heuristic condition" procedure (see 11.5.15).

c) If this is an AF-ABORT-AND-REPORT indication and the Completion Diagnostics functional unit is selected on this (subordinate) dialogue and any of the Severity, Diagnostic or Completion-data parameters are present:

   – issue a TP-COMPLETION-REPORT indication, with the Completion-data parameter set to the value of the Completion-data parameter of the AF-ABORT-AND-REPORT indication.

d) If a TP-U-ABORT request was received:

   – continue.

e) If no TP-U-ABORT request was received:

   – issue a TP-U-ABORT indication with the Rollback parameter set to "false" if the value of the Type parameter is "user" or this is an AF-ABORT-AND-REPORT indication;

   – issue a TP-P-ABORT indication with the Rollback parameter set to "false" if the value of the Type parameter is "provider".

f) If the *last rollback confirm was received* and this is an intermediate node and the superior *dialogue has not been detached*:

   – invoke the "Reporting rollback to superior" procedure (see 11.5.16).

g) If the indication was received from a subordinate, the *last rollback confirm was received*, and the *superior dialogue is not chaining* and *rollback reporting has completed*:

   – invoke the "Initiating transaction after rollback" procedure (see 11.5.11).

h) If the indication was received from the superior and *a TP-DONE request is not owed*:

   – invoke the "Initiating transaction after rollback" procedure (see 11.5.11).

If the TPPM is in the EARLY-EXIT state, this is an AF-ABORT indication on the superior dialogue and the Type parameter is set to "user":

– issue a TP-U-ABORT indication with the Rollback parameter set to "true";

– invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "superior-rollback";

– issue an SAF-DETACH-ASSOCIATION (free) request.

11.3.59 AF-REPORT (dataRI) indication or AF-ABORT-AND-REPORT (dataRI) indication

The primitive shall be on a subordinate dialogue.

If *heuristic reporting applies* and the Heuristic Report parameter is not set to "none":

– issue a TP-HEURISTIC-REPORT indication;

– invoke the "Recording the heuristic condition" procedure (see 11.5.15).

If the Completion diagnostics functional unit is selected on this dialogue and any of the Severity, Diagnostic or Completion-data parameters are present:

– issue a TP-COMPLETION-REPORT indication.

If the *reporting status is known* and *reporting applies* on the branch to the superior:

– invoke the "Reporting on the commit-coordinator:root path" procedure (see 11.5.4).

NOTE – If a TP-COMPLETION-REPORT indication was issued, a *TP-DONE is owed*, and so the *reporting status will not be known*. If no TP-COMPLETION-REPORT indication was issued, it is possible the *reporting status is known.*

If the *last commit confirm* has been received and the *reporting status is known* and the TPPM is in the DECIDED (commit) state:

– invoke the "Confirm and completing commitment" procedure (see 11.5.1).

If the TPPM is in the DECIDED (commit-one-phase) state and a TP-DONE is not owed:

– issue an AF-ABORT (user, dataRI) request, if a TP-U-ABORT request was received and no AF-DEFERRED-END-DIALOGUE request was issued and this is an AF-REPORT indication;

– issue an SAF-DETACH-ASSOCIATION (free) request if TP-U-ABORT request was received or an AF-DEFERRED-END-DIALOGUE request was issued or this is an AF-ABORT-AND-REPORT indication (or any combination of these three are true);

– invoke the "Completing ONE-PHASE and READ-ONLY" procedure (see 11.5.5).

If none of the above conditions is true:

– continue.

11.3.60 TP-ONE-PHASE request

For each subordinate to which an AF-PREPARE request has not already been issued:

a) If an TP-DEFERRED-END-DIALOGUE request was received and no AF-DEFER (end-dialogue) request was issued:

   – issue an AF-DEFER (end-dialogue) request.

b) If no TP-DEFERRED-END-DIALOGUE request was received, a TP-DEFERRED-GRANT-CONTROL request was received and no AF-DEFER (grant-control) request was issued:

   – issue an AF-DEFER (grant-control) request.

If the TPPM is in the ACTIVE state, perform the first applicable of the following sets of actions:

a) If the *last ready was received*:

   – invoke the "Making commitment decision" procedure (see 11.5.12).

b) If the *ready state may be entered*:

   – invoke the "Entering READY state" procedure (see 11.5.6).

c) If the *last ready-substitute has been received*:

   – invoke the "Making one-phase commitment decision" procedure (see 11.5.13).

d) If the *non-recovery states may be entered*:

   – invoke the "Entering ONE-PHASE or READ-ONLY state" procedure (see 11.5.20).

e) Otherwise, for each dialogue supporting a branch of the transaction on which no AF-PREPARE request has been issued and no C-READY indication or *ready-substitute indication* has been received; and at least one of the following is true:

   i) *ready is receivable* and the dialogue is to a subordinate and the Implicit Prepare functional unit is not selected; or

   ii) *ready is receivable* and the Polarized Control functional unit is selected and an AF-PREPARE (no-data-permitted) indication has not been received; or

iii) *a local decision determines*.

NOTE – It is expected this local decision will be configurable for each dialogue. The effect of iii) begin true is equivalent to combining, as a single action, the *completion request* with TP-PREPARE requests for the dialogues for which i) and ii) are false.

Then, on that dialogue:

– issue an AF-PREPARE request with the Data-Permitted parameter:

1) absent, if the Shared Control functional unit is selected; or

2) set to "false", if the Polarized Control functional unit is selected.

11.3.61 TP-READ-ONLY request

The Read Only functional unit shall be selected on the superior dialogue if there is one.

For each subordinate to which an AF-PREPARE request has not already been issued:

a) If an TP-DEFERRED-END-DIALOGUE request was received and no AF-DEFER (end-dialogue) request was issued:

– issue an AF-DEFER (end-dialogue) request.

b) If no TP-DEFERRED-END-DIALOGUE request was received, a TP-DEFERRED-GRANT-CONTROL request was received and no AF-DEFER (grant-control) request was issued:

– issue an AF-DEFER (grant-control) request.

If the TPPM is in the ACTIVE state, perform the first applicable of the following sets of actions:

a) If the *last ready has been received*:

– invoke the "Making commitment decision" procedure (see 11.5.12).

b) If the *ready state may be entered*:

– invoke the "Entering READY state" procedure (see 11.5.6).

c) If the *last ready-substitute has been received*:

– invoke the "Making one-phase commitment decision" procedure (see 11.5.13).

d) If the *non-recovery states may be entered*:

– invoke the "Entering ONE-PHASE or READ-ONLY state" procedure (see 11.5.20).

e) Otherwise, for each dialogue supporting a branch of the transaction on which no AF-PREPARE request has been issued and no C-READY indication or *ready-substitute indication* has been received, and at least one of the following is true:

i) *ready is receivable; and* the dialogue is to a subordinate and the Implicit Prepare functional unit is not selected; or

ii) *ready is receivable*; and the Polarized Control functional unit is selected and an AF-PREPARE (no-data-permitted) indication has not been received; or

iii) *a local decision determines*.

NOTE – It is expected this local decision will be configurable for each dialogue. The effect of iii) begin true is equivalent to combining, as a single action, the *completion request* with TP-PREPARE requests for the dialogues for which i) and ii) are false.

Then:

– issue an AF-PREPARE request with the Data-Permitted parameter:

i) absent, if the Shared Control functional unit is selected; or

ii) set to "false", if the Polarized Control functional unit is selected.

11.3.62 AF-NOCHANGE indication or C-NOCHANGE indication

If this is a subordinate dialogue, the *dialogue is not chaining,* no AF-DEFER request has been issued on the dialogue, the Confirmation parameter is "result-not-required", the TPPM is not in a *user error purging period* and a handshake request is not outstanding:

– issue a C-NOCHANGE response with the Outcome parameter set to "not-determined".

NOTE 1 – The coordination level now becomes "none", and the dialogue is no longer part of the transaction.

If the TPPM is in the ACTIVE state, perform the first applicable of the following sets of actions:

a) If the TPPM is in a *user error purging period* or a handshake request is outstanding:

NOTE 2 – A handshake request outstanding may occur only if the Shared Control functional unit was selected; a C-READY indication can only be received under either of these conditions if at least one of the Dynamic Commit and Implicit Prepare functional units were selected.

- invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "user-data-transaction-completion-collision".

b) If no *completion request* has been received and either this is a superior dialogue or it is a subordinate dialogue and the Confirmation parameter is "result-requested":

- issue a TP-ONE-PHASE indication.

c) If no *completion request* has been received, this is a subordinate dialogue and the Confirmation parameter is not "result-requested:

- issue a TP-READ-ONLY indication;
- cease to be part of the transaction; if this is a root node, the dialogue *is not chaining* and there are no other coordinated dialogues;
- become a leaf node; if this is an intermediate, the dialogue *is not chaining* and there are no other coordinated subordinate dialogues.

d) If the *last ready was received*:

- invoke the "Making commitment decision" procedure (see 11.5.12).

e) If the *ready state may be entered*:

- invoke the "Entering READY state" procedure (see 11.5.6).

f) If the *last ready-substitute has been received*:

- invoke the "Making one-phase commitment decision" procedure (see 11.5.13).

g) If the *non-recovery states may be entered*:

- invoke the "Entering ONE-PHASE or READ-ONLY state" procedure (see 11.5.20).

h) Otherwise:

- continue.

If the TPPM is in the DECIDED (rollback) state:

- continue.

If the TPPM is in the READY state:

- invoke the "Making commitment decision" procedure (see 11.5.12).

NOTE 3 – This can only occur if there is a READY/ONE-PHASE collision. This node becomes the commit-coordinator, rewriting its log.

11.3.63 TP-EARLY-EXIT request

The Early-exit functional unit shall be selected on the superior dialogue.

If the TPPM is in the ACTIVE state, perform the first applicable of the following sets of actions:

a) If there is TPPM bound data:

- invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "local-rollback".

b) Otherwise:

- issue an AF-EARLY-EXIT request on the superior dialogue;
- enter the EARLY-EXIT state.

11.3.64 AF-EARLY-EXIT indication

The dialogue shall be from a subordinate.

Perform the first applicable of the following sets of actions:

a) If the TPPM is in the ACTIVE state and no *completion-request* has been received:
   - issue an AF-EARLY-EXIT response;
   - issue a TP-EARLY-EXIT indication;
   - cease to be part of the transaction; if this is a root node, the dialogue *is not chaining* and there are no other coordinated dialogues;
   - become a leaf node, if this is an intermediate, the dialogue *is not chaining* and there are no other coordinated subordinate dialogues.

b) If the TPPM is in the DECIDED (rollback) state:
   1) if the Severity or Completion Data parameters are present:
      - issue a TP-COMPLETION-REPORT indication;
   2) if a TP-U-ABORT request was received:
      - issue an AF-ABORT (user, rollbackRC) request;
      - issue an SAF-DETACH-ASSOCIATION (free) request;
   3) if a TP-U-ABORT request was not received:
      - issue a C-ROLLBACK response;
      - *close the PSAP*, if the Unchained Transactions functional unit is selected on this dialogue;
   4) if the *last rollback confirm was received*, this is an intermediate node, and the *superior dialogue has not been detached*:
      - invoke the "Reporting rollback to superior" procedure (see 11.5.16);
   5) if the *last rollback confirm was received* and the *superior dialogue is not chaining* and *rollback reporting has completed*:
      - invoke the "Initiating transaction after rollback" procedure (see 11.5.11).

c) Otherwise:
   - invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "early-exit-transaction-completion-collision";
   - issue a TP-COMPLETION-REPORT indication if the Severity or Completion-data parameters are present and non-empty.

11.3.65 AF-EARLY-EXIT confirm

The dialogue shall be from the superior and the TPPM shall be in the EARLY-EXIT state.

Always:

- enter the DECIDED (unknown) state;
- issue a TP-UNKNOWN indication.

If a TP-U-ABORT request was received:

- issue an AF-ABORT (user, dataRI) request;
- issue an SAF-DETACH-ASSOCIATION (begin-fear) request.

If no TP-U-ABORT request was received:

- *close the PSAP*.

If a C-NOCHANGE indication has been received on any other dialogue that is still in the transaction:

- invoke the "Sending not-determined result from a ONE-PHASE or READ-ONLY node" procedure (see 11.5.19).

11.3.66 CAF-RECOVER (ready) indication

If the TPPM is in the:

a) READY state, and:

   1) the *ready signal has been sent* to this neighbour:

      – invoke the "Initiating rollback at TPPM" procedure (see 11.5.10);

   2) the *ready signal was not sent* to this neighbour and:

      i) if no other channel for this branch identifier already exists, by a local decision, either:

         – issue a C-RECOVER (retry-later) response;

         – issue a CAF-DETACH (free) request; or

         – continue;

         NOTE 1 – The C-RECOVER (retry-later) response may be issued to release the channel while waiting for the *commit indication* from the superior.

      ii) if another channel for this branch identifier already exists, by a local decision, do either or both of:

         – issue a C-RECOVER (retry-later) response on the previously existing channel; and

         – issue a CAF-DETACH (free) request on the previously existing channel; or:

         – issue a C-RECOVER (retry-later) response on the channel from which this indication was received; and

         – issue a CAF-DETACH (free) request on the channel from which this indication was received.

         NOTE 2 – This situation will arise when one of the channels has been aborted, but the abort has not yet been signalled to the TPPM. This Recommendation does not provide enough information to determine which channel has been aborted, so it is left to an implementation to make the choice of the channel to be released since the implementation may have local information to help determine which channel has been aborted. In addition, a local decision may be made to issue C-RECOVER (retry-later) responses and CAF-DETACH (free) requests on both channels while waiting for the commit indication from the superior.

b) DECIDED (commit) state:

   1) if no channel exists and a commit confirm was not received on either the dialogue or a channel for this branch identifier:

      – issue an AF-REPORT (recoverCommitRI) request *carrying the reporting status* if the branch is to a superior and the *reporting status is known* and *reporting applies* on this branch;

      – issue a C-RECOVER (commit) request otherwise;

   2) if a channel exists or a commit confirm was received on either the dialogue or a channel for this branch identifier:

      – issue a C-RECOVER (retry-later) response on the channel from which this indication was received;

      – issue a CAF-DETACH (free) request on the channel from which this indication was received.

c) DECIDED (rollback) state or the DECIDED (unknown) state:

   – issue a C-RECOVER (unknown) response;

   – issue a CAF-DETACH (free) request.

If the *dialogue* corresponding to the value contained in the Branch Identifier parameter *has not been detached*, and the TPPM is in the:

a) ACTIVE state (provided an AF-PREPARE request was sent or the Implicit Prepare functional unit is selected on the dialogue or the Dynamic Commit functional unit is selected on the dialogue which is to the superior), ONE-PHASE or READ-ONLY state:

   – issue a TP-P-ABORT indication with the:

      1) Diagnostic parameter set to "permanent-failure"; and

      2) Rollback parameter set to "true";

- issue an AF-ABORT (provider, abortRI) request on the dialogue, with the Diagnostic parameter set to "permanent-failure";
- issue a C-RECOVER (unknown) response on the channel;
- issue a CAF-DETACH (free) request on the channel;
- invoke the "Initiating rollback at TPPM" procedure (see 11.5.10);
- issue a TP-HEURISTIC-REPORT indication with the Heuristic-Report parameter set to "heuristic-hazard", unless the dialogue was to a subordinate or the Heuristic Containment Required functional unit was selected on the dialogue;
- invoke the "Recording the heuristic condition" procedure (see 11.5.15), unless the dialogue was to a subordinate.

b) READY, DECIDED (commit), or DECIDED (rollback) state:

- issue a TP-P-ABORT indication if no TP-U-ABORT request was received, with the:

  1) Diagnostic parameter set to "permanent-failure"; and

  2) Rollback parameter set to "false";

- issue an AF-ABORT (provider, abortRI) request on the dialogue.

c) DECIDED (rollback) state:

- issue a TP-HEURISTIC-REPORT indication with the Heuristic-Report parameter set to "heuristic-hazard", unless the dialogue was to a subordinate or the Heuristic Containment Required functional unit was selected on the dialogue;
- invoke the "Recording the heuristic condition" procedure (see 11.5.15), unless the dialogue was to a subordinate or the Heuristic Containment Required functional unit was selected on the dialogue.

d) DECIDED (commit) state:

- invoke the "Rollback next transaction" procedure (see 11.5.17) if a C-COMMIT+C-BEGIN request was issued.

11.3.67 C-RECOVER (ready) indication or AF-RECOVER (ready) indication (CPM)

If this is an AF-RECOVER (ready) indication, according to a local decision, either:

- issue a C-RECOVER (retry-later) response; or

  NOTE 1 – A C-RECOVER (retry-later) response may be issued if it is currently impossible to determine whether the log record exists [e.g. the portion of the set of log records identified by the recovery-context-handle (see 7.5) is currently inaccessible].

- attempt to locate a TPPM having a subordinate branch with an atomic action identifier, branch identifier, and recovery-context-handle corresponding to the parameters of the AF-RECOVER (ready) indication.

If this is a C-RECOVER (ready) indication, according to a local decision, either:

- issue a C-RECOVER (retry-later) response; or

  NOTE 2 – A C-RECOVER (retry-later) response may be issued if it is currently impossible to determine whether the log record exists [e.g. the portion of the set of log records identified by the absence of a recovery-context-handle (see 7.5) is currently inaccessible].

- attempt to locate a TPPM having a subordinate branch with an atomic action identifier and branch identifier, corresponding to the parameters of the C-RECOVER (ready) indication.

If a TPPM is found:

- invoke the "Fail an outstanding CAF-PLEASE request" procedure (see 11.5.7), if a *CAF-PLEASE request is outstanding* for the channel;
- issue a CAF-RECOVER (ready) indication to the TPPM that has been found. Set the corresponding parameters of the CAF-RECOVER (ready) indication from the C-RECOVER (ready) indication or AF-RECOVER (ready) indication and set the Channel-Utilization parameter to that of the most recent AF-BEGIN-DIALOGUE request or indication on this channel.

If no TPPM is found and a C-RECOVER (retry-later) response was not issued:

- issue a C-RECOVER (unknown) response.

11.3.68 C-NOCHANGE confirm or AF-ABORT (user, nochangeRC) indication

The TPPM shall be in the ONE-PHASE or READ-ONLY state.

If the TPPM is in the ONE-PHASE state and the Outcome parameter of primitive has the value "commit" or "no-change":

- enter the DECIDED (commit-one-phase) state;
- issue a TP-COMMIT indication;
- invoke the "Sending commit order" procedure (see 11.5.18), if a C-NOCHANGE indication was received on any other dialogue.

If the TPPM is in the READ-ONLY state or the Outcome parameter of primitive has the value "not-determined":

- enter the DECIDED (unknown) state;
- issue a TP-UNKNOWN indication;
- invoke the "Sending not-determined result from a ONE-PHASE or READ-ONLY node" procedure (see 11.5.19), if a C-NOCHANGE indication was received on any other dialogue.

If this is an AF-ABORT (user, nochangeRC) indication:

- issue a TP-U-ABORT indication with the Rollback parameter set to "false", if no TP-U-ABORT request has been received for the dialogue;
- issue an SAF-DETACH-ASSOCIATION (free) request if this is a subordinate dialogue, or a superior dialogue and the Chained functional unit is selected;
- issue an SAF-DETACH-ASSOCIATION (begin-fear) request if this is a superior dialogue and the Unchained functional unit is selected.

If this is a C-NOCHANGE confirm on a superior dialogue and:

1) If a TP-U-ABORT request was received:
   - issue an AF-ABORT (user, dataRI) request;
   - issue an SAF-DETACH-ASSOCIATION (begin-fear) request.
2) If no TP-U-ABORT request was received and an AF-DEFER (end-dialogue) indication was received:
   - issue an SAF-DETACH-ASSOCIATION (free) request.
3) If neither of the above conditions apply:
   - continue.

If this is a C-NOCHANGE confirm on a subordinate dialogue and a TP-DEFERRED-END-DIALOGUE request was received and no TP-U-ABORT request was received and *reporting does not apply* on the dialogue:

- issue an SAF-DETACH-ASSOCIATION (free) request.

If this is a C-NOCHANGE confirm on a subordinate dialogue and a TP-U-ABORT request was received and *reporting does not apply* on the dialogue:

- issue an AF-ABORT (user, dataRI) request, if no AF-DEFERRED-END-DIALOGUE request was issued;
- issue an SAF-DETACH-ASSOCIATION (free) request.

11.3.69 CAF-RECOVER (commit) indication

If the TPPM is in the READY state:

- invoke the "Receiving commit order" procedure (see 11.5.14);
- invoke the "Sending commit order" procedure (see 11.5.18) if a *ready signal was received* on any other branch.

If the primitive is from a subordinate, *reporting applies on the branch,* a TP-HEURISTIC-REPORT indication has not been issued and the Heuristic-Report parameter is present and does not have the value "none":

- issue a TP-HEURISTIC-REPORT indication;
- invoke the "Recording the heuristic condition" procedure (see 11.5.15).

NOTE 1 – Issue of a TP-HEURISTIC-REPORT indication prior to a node crash will not be remembered [see 7.2 h)].

NOTE 2 – A TP-HEURISTIC-REPORT could have been issued as a result of an AF-REPORT indication on the original dialogue or a previous CAF-RECOVER indication or C-RECOVER indication.

If the primitive is from a subordinate and *reporting applies on the branch* and the Heuristic-Report parameter is absent:

a) Always:

   – issue a C-RECOVER (retry-later) response;

   – issue a CAF-DETACH (free) request.

   NOTE 3 – The heuristic status will be reported on a subsequent CAF-RECOVER or AF-REPORT (recoverRI) indication.

b) If a channel for this branch identifier already exists and the TPPM is in the READY state:

   – issue a CAF-DETACH (clean-up) request on the previously existing channel.

If the primitive is from the superior, or from a subordinate and *reporting does not apply on the branch* or from a subordinate and the Heuristic-Report parameter is present, then:

a) If the TPPM is in the READY state, or DECIDED (commit) state and the *last commit confirm was not received* and an *intermediate log-record has not been rewritten*:

   1) if no channel for this branch identifier already exists and, by a local decision, either:

      – issue a C-RECOVER (retry-later) response; and

      – issue a CAF-DETACH (free) request; or

      – continue.

      NOTE 4 – The C-RECOVER (retry-later) response may be issued to release the channel while waiting for the *commit confirms*.

   2) If a channel for this branch identifier already exists and the TPPM is in the READY state, by a local decision, either:

      – issue a CAF-DETACH (clean-up) request on the previously existing channel;

      – issue a C-RECOVER (retry-later) response on the channel from which this indication was received; and

      – issue a CAF-DETACH (free) request on the channel from which this indication was received; or

      – issue a CAF-DETACH (clean-up) request on the previously existing channel.

      NOTE 5 – The C-RECOVER (retry-later) response may be issued to release the channel while waiting for the *commit confirms* from the subordinates. In any case, the previously existing channel is *detached* because the TPPM now knows the outcome of the transaction is commit, so the reply from the previous channel is uninteresting.

   3) If a channel for this branch identifier already exists and the TPPM is in the DECIDED (commit) state, by a local decision, do either or both of:

      – issue a C-RECOVER (retry-later) response on the previously existing channel; and

      – issue a CAF-DETACH (free) request on the previously existing channel; or

      – issue a C-RECOVER (retry-later) response on the channel from which this indication was received; and

      – issue a CAF-DETACH (free) request on the channel rom which this indication was received.

      NOTE 6 – This situation will arise when one of the channels has been aborted, but the abort has not yet been signalled to the TPPM. This Recommendation does not provide enough information to determine which channel has been aborted, so it is left to an implementation to make the choice of the channel to be released since the implementation may have local information to help determine which channel has been aborted. In addition, a local decision may be made to *detach* both channels while waiting for the *commit confirms* from the subordinates.

b) If the TPPM is in the DECIDED (commit) state and an *intermediate log-record has been rewritten*:

   – issue a C-RECOVER (done) response; and

   – issue a CAF-DETACH (free) request.

If the *dialogue* corresponding to the value contained in the Branch Identifier parameter *has not been detached*:

- issue a TP-P-ABORT indication if no TP-U-ABORT request was issued, with the:
  1) Diagnostic parameter set to "permanent-failure"; and
  2) Rollback parameter set to "false";
- issue an AF-ABORT (provider, abortRI) request on the dialogue, with the Diagnostic parameter set to "permanent-failure";
- invoke the "Rollback next transaction" procedure (see 11.5.17) if a C-COMMIT+C-BEGIN request was received on the dialogue.

11.3.70 C-RECOVER (commit) indication or AF-REPORT (recoverCommitRI) indication

The TPPM shall be in the READY state:

- invoke the "Receiving commit order" procedure (see 11.5.14);
- invoke the "Sending commit order" procedure (see 11.5.18) if this node is an intermediate.

If this is an AF-REPORT indication, and the Heuristic-Report parameter does not have the value "none":

- issue a TP-HEURISTIC-REPORT indication;
- invoke the "Recording the heuristic condition" procedure (see 11.5.15).

If this is a C-RECOVER (commit) indication from a subordinate, and *reporting applies on the branch*:

- issue a C-RECOVER (retry-later) response;
- issue a CAF-DETACH (free) request.

NOTE 1 – The heuristic status will be reported on a subsequent CAF-RECOVER or AF-REPORT (recoverCommitRI) indication.

By a local decision, optionally, if this is from the superior or is an AF-REPORT indication or *reporting does not apply on the branch*:

- issue a C-RECOVER (retry-later) response;
- issue a CAF-DETACH (free) request.

NOTE 2 – The C-RECOVER (retry-later) response may be issued to release the channel while waiting for the *commit confirms* from the subordinates.

11.3.71 C-RECOVER (commit) indication or AF-RECOVER (commit) indication or AF-REPORT (recoverCommitRI) indication (CPM)

If the most recent request from the TPPM on this channel was a CAF-DETACH (clean-up) request and this is a C-RECOVER (commit) indication:

- issue a C-RECOVER (retry-later) response.

If the most recent request from the TPPM on this channel was not a CAF-DETACH (clean-up) request:

a) If this is an AF-RECOVER (commit) indication or an AF-REPORT indication on which the Recovery-Context-Handle is present, according to a local decision, either:

   - issue a C-RECOVER (retry-later) response; or

     NOTE 1 – A C-RECOVER (retry-later) response may be issued if it is currently impossible to determine whether the log record exists [e.g. the portion of the set of log records identified by the recovery-context-handle (see 7.5) is currently inaccessible].

   - attempt to locate a TPPM having a superior branch with an atomic action identifier, branch identifier, and recovery-context-handle corresponding to the parameters of the AF-RECOVER (commit) indication or AF-REPORT indication.

b) If this is a C-RECOVER (commit) indication or an AF-REPORT indication on which the Recovery-Context-Handle is absent, according to a local decision, either:

   - issue a C-RECOVER (retry-later) response; or

     NOTE 2 – A C-RECOVER (retry-later) response may be issued if it is currently impossible to determine whether the log record exists [e.g. the portion of the set of log records identified by the absence of a recovery-context-handle (see 7.5) is currently inaccessible].

- attempt to locate a TPPM having a superior branch with an atomic action identifier and branch identifier, corresponding to the parameters of the C-RECOVER (commit) indication or AF-REPORT indication.

c) If a TPPM is found:

   - invoke the "Fail an outstanding CAF-PLEASE request" procedure (see 11.5.7), if a *CAF-PLEASE request is outstanding* for this channel;
   - issue a CAF-RECOVER (commit) indication to the TPPM that has been found. Set the corresponding parameters of the CAF-RECOVER (commit) indication from the C-RECOVER (commit) indication or AF-RECOVER (commit) indication or AF-REPORT indication and set the channel-utilization parameter to that of the most recent AF-BEGIN-DIALOGUE request or indication on this channel; if this was an AF-REPORT indication, set the Heuristic-Report parameter on the CAF-RECOVER indication from that parameter of the AF-REPORT; otherwise the Heuristic-Report parameter shall be absent.

d) If no TPPM is found and a C-RECOVER (retry-later) response was not issued:

   - issue a C-RECOVER (done) response if no log-damage record exists;
   - issue an AF-REPORT (recoverDoneRC) request if a log-damage record exists; in this case, the Heuristic-Report parameter is set to the current value of the log-damage record.

11.3.72 C-RECOVER (done) confirm or AF-REPORT (recoverDoneRC) indication

If this is an AF-REPORT indication:

- issue a TP-HEURISTIC-REPORT indication;
- invoke the "Recording the heuristic condition" procedure (see 11.5.15).

If a *commit-request* was sent to the superior and the *reporting status is known* and *reporting applies* on the branch to the superior and *reports have not been sent* to the superior:

- invoke the "Reporting on the commit-coordinator:root path" procedure (see 11.5.4).

If the last *commit confirm* was received and the *reporting status is known*:

- invoke the "Confirm and complete commitment" procedure (see 11.5.1).

Always:

- issue a CAF-DETACH (free) request.

11.3.73 C-RECOVER (unknown) confirm

- Issue a CAF-DETACH (free) request;
- invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) if the TPPM is in the READY state. The diagnostic value shall be "superior-rollback" if the primitive is from the superior, "subordinate-rollback" if the primitive is from a subordinate.

11.3.74 C-RECOVER (unknown) confirm (CPM)

If the most recent request from the TPPM on this channel was a CAF-DETACH (clean-up) request:

- continue.

11.3.75 C-RECOVER (retry-later) confirm

- Issue a CAF-DETACH (free) request.

NOTE – The requirement to perform recovery is not absolved. The internal event "Retry recovery" (see 11.4.4) is used to initiate recovery, or recovery could be initiated by the partner, or both. If this primitive is in reply to a C-RECOVER (commit) request issued to the superior (using Dynamic Commit) and *reporting applies on the branch* but the *reporting status is not known*, a recovery attempt will be initiated when the reporting status does become known. Implementations can use this fact to modify the scheduling of retry attempts.

11.3.76 C-RECOVER (retry-later) confirm (CPM)

If the most recent request from the TPPM on this channel was a CAF-DETACH (clean-up) request:

- continue.

11.3.77 AF-TOKEN-GIVE (two-way-recovery) indication on a channel (TPPM)

If this is a two-way-recovery channel and the two-way-recovery token is expected:

– continue.

11.3.78 AF-TOKEN-GIVE (two-way-recovery indication (CPM)

If this is a two-way-recovery channel, *a CAF-PLEASE request is outstanding* for the channel, and a TPPM with a branch corresponding to the "Atomic Action Identifier", "Branch Identifier", and "Superior" parameters specified in the previous CAF-PLEASE request is found:

– issue a CAF-GIVE indication to the requesting TPPM. Set the Channel-Utilization parameter to "two-way-recovery";

otherwise, if this is a two-way-recovery channel:

– continue.

11.3.79 AF-TOKEN-PLEASE indication on a channel (TPPM)

If this is a two-way-recovery channel:

– continue.

NOTE – The receipt of an AF-TOKEN-PLEASE indication by the TPPM will always be the result of a collision of an AF-TOKEN-PLEASE request issued by a CPM and a C-RECOVER request or AF-RECOVER request issued by the TPPM.

11.3.80 AF-TOKEN-PLEASE indication (CPM)

If this is a two-way-recovery channel:

– issue an AF-TOKEN-GIVE (two-way-recovery) request.

11.3.81 CAF-PLEASE request (CPM)

If there is already an established channel owned by the CPM to the appropriate AEI, which is a two-way-recovery channel or a one-way-recovery channel initiated by the CPM, and:

a) If this is a:

   1) one-way-recovery channel;

   2) two-way-recovery channel and the *token* is owned; or

   3) two-way-recovery channel and the last request received on the channel was a CAF-DETACH (not-used) request,

   then:

   – *transfer the channel* to the requesting TPPM;

   – issue a CAF-GIVE indication to the requesting TPPM.

b) If this is a two-way-recovery channel, the *token* is not owned, and the last request received on the channel was not a CAF-DETACH (not-used) request:

   – issue an AF-TOKEN-PLEASE request, if *the two-way-recovery token is not expected* and *an AF-TOKEN-PLEASE request is not outstanding*;

   – continue,

otherwise:

– assign an association compatible with the AEI requirements for a channel and to the AE-Title as specified on the CAF-PLEASE request;

– issue an AF-BEGIN-DIALOGUE request with the value of the:

   - Functional-Units-Selected parameter set to "Recovery";

   - Channel-Utilization parameter set to either "one-way-recovery" or "two-way-recovery" based on a local decision.

11.3.82 CAF-GIVE indication

If the TPPM is in the:

a) READY state, and no channel is *attached*:

- issue a C-RECOVER (ready) request if no recovery-context-handle is present in the log-ready record;
- issue an AF-RECOVER (ready) request if a recovery-context-handle is present in the log-ready record;
- issue an AF-TOKEN-GIVE (two-way-recovery) request if this is a two-way-recovery channel.

b) DECIDED (commit) state and the recovery is to a neighbour from whom a *commit indication was received*:

- issue a CAF-DETACH (not-used) request.

c) DECIDED (commit) state, the recovery is to a neighbour from whom a *ready signal was received*, no channel is *attached*, a C-RECOVER (done) confirm was not received and the recovery is to a subordinate, or to the superior and *reporting does not apply* or the *reporting status is not known*:

- issue a C-RECOVER (commit) request if no recovery-context-handle is present in the log-commit record;
- issue an AF-RECOVER (commit) request if a recovery-context-handle is present in the log-commit record;
- issue an AF-TOKEN-GIVE (two-way-recovery) request if this is a two-way-recovery channel.

d) DECIDED (commit) state, the recovery is to the superior from whom a *ready signal was received*, no channel is *attached*, a C-RECOVER (done) confirm was not received, the *reporting status is known* and *reporting applies* on the branch:

- issue an AF-REPORT (recoverCommitRI) request;
- issue an AF-TOKEN-GIVE (two-way-recovery) request if this is a two-way-recovery channel.

e) DECIDED (commit) state, the recovery is to a neighbour from whom a *ready signal was received*, and either a channel is *attached* or a C-RECOVER (done) was received:

- issue a CAF-DETACH (not-used) request.

NOTE – Between the time the TPPM requested the channel for recovering a particular branch, an incoming recovery indication for that branch might have occurred, removing the need for a channel.

11.3.83 CAF-FAIL indication

If the TPPM is in the:

a) READY state, and no channel is *attached*:

- issue a CAF-PLEASE request with the AE-Title of the CAF-PLEASE request set to the value of the AE-Title taken from the branch identifier for the neighbour to which a *ready signal was sent*, contained in the log-ready record.

b) DECIDED (commit) state and the indication is from a neighbour from whom a *commit indication was received*:

- continue.

c) DECIDED (commit) state, the recovery is to a neighbour from whom a *ready signal was received*, no channel is *attached*, and a C-RECOVER (done) confirm was not received:

- issue a CAF-PLEASE request with the AE-Title of the CAF-PLEASE request set to the value of the AE-Title taken from the branch identifier for the neighbour, contained in the log-ready record.

d) DECIDED (commit) state, the recovery is to a neighbour from whom a *ready signal was received*, and either a channel is *attached* or a C-RECOVER (done) confirm was received:

- continue.

11.3.84 CAF-DETACH request (CPM)

- *Transfer the channel* to the CPM.

11.4 Internal event procedures

These procedures are invoked upon the receipt of an event local to a PM. The first paragraph of each procedure describes the conditions under which the procedure is invoked.

11.4.1 Delay recovery

This procedure is invoked when the TPPM will release the channel on which an AF-RECOVER indication or a C-RECOVER indication but no corresponding AF-RECOVER response or C-RECOVER response was issued:

– issue a C-RECOVER (retry-later) response;

– issue a CAF-DETACH (free) request.

NOTE 1 – In the absence of a TPPM state change, it is the partner's responsibility to initiate recovery. In the situation where the TPPM state changes from READY state to DECIDED (commit) state, the TPPM will initiate recovery.

NOTE 2 – This is used when it is undesirable for the TPPM to hold the channel while waiting for the conditions necessary to issue the C-RECOVER response.

11.4.2 Heuristic damage compensation for subtree

This procedure is invoked when it is known that heuristic damage compensation has occurred for the subtree. It causes the log-damage and log-heuristic records to be deleted since the damage was corrected. This prevents the reporting of this damage to the superior:

– remove the log-damage record from secure storage.

If a log-heuristic record exists:

– remove the log-heuristic record from secure storage.

11.4.3 Restart after node crash (CPM)

The PM procedures assume that this procedure is entered before any other procedure after a node crash.

The MACF of a TPPM which cannot continue normal operation is deleted. The SAOs which were part of this TPPM have disappeared by an association abort or they are in a state such that arriving APDUs cannot interfere with the operation of the TPPM re-established after the node crash.

The CPM instantiates a TPPM for each log-ready record or log-commit record found, provided a TPPM within the AEI does not already exist for the log record, as follows:

NOTE – For each TPPM, the internal event "TPPM creation after node crash" (see 11.4.7) will occur.

a) If a log-commit record was found:

   – create the TPPM in the DECIDED (commit) state.

b) If a log-ready record was found:

   – create the TPPM in the READY state.

For all *TPPM bound data* for which no log record is found:

– set the *TPPM bound data* to the initial state.

11.4.4 Retry recovery

This procedure may be invoked at any time while the TPPM's responsibility to perform recovery is not yet absolved, and the TPPM has not issued a CAF-PLEASE request nor is it processing recovery of the branch.

NOTE – A timer is a typical mechanism used to invoke this procedure.

– Issue a CAF-PLEASE request with the AE-Title of the CAF-PLEASE request set to the value of the AE-Title taken from the branch identifier for the superior, contained in the log-ready record, if this is a branch to the superior, or set to the value of the AE-Title taken from the branch identifier for the subordinate, contained in the log-commit record, if this is a branch to a subordinate.

11.4.5 Taking a heuristic decision

This procedure is entered when the TPPM takes a heuristic decision.

A TPPM may take a heuristic decision only while in the ACTIVE state if an AF-PREPARE indication has been received, or while in the READY state:

– part or all of the bound data is set to the initial, or final state;

– secure the heuristic decision by *writing the log-heuristic record*.

11.4.6 Terminating a channel (CPM)

This procedure is entered when the CPM terminates a channel according to a local decision.

If the *token* is owned and the channel is *attached* to the CPM:

– issue an AF-END-DIALOGUE request with the Confirmation parameter set to "false";

– issue an SAF-DETACH-ASSOCIATION (free) request.

11.4.7 TPPM creation after node crash

This procedure is invoked as the first one in a newly created TPPM after a node crash. This procedure is executed as an integral part of the creation of the TPPM.

NOTE – This internal event procedure is entered immediately after completion of the internal event procedure "Restart after node crash (CPM)", if a TPPM was created.

– Issue a TP-COMMIT indication if the TPPM is in the DECIDED (commit) state.

If the TPPM is in the DECIDED (commit) state then, for each neighbour in the log-commit record from which a *ready signal was received*:

– issue a CAF-PLEASE request with the AE-title, Atomic Action Identifier, Branch Identifier and Recovery-Context-Handle parameters set from the values for that neighbour in the *log-commit record.*

If the TPPM is in the READY state, for the neighbour to which a *ready signal was sent*:

– issue a CAF-PLEASE request with the AE-title, Atomic Action Identifier, Branch Identifier and Recovery-Context-Handle parameters set from the values for that neighbour in the *log-ready record.*

11.4.8 TPPM-initiated rollback

This procedure may be invoked without any preconditions in ACTIVE state except when a dialogue establishment indication is outstanding:

– invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic value of "local-rollback".

NOTE – Deadlocks occurring due to a node being unable to progress to a READY state, after issue of *completion request* will typically require a TPPM-initiated rollback. This can normally be detected locally. If both Dynamic Commit and Unchecked Tree functional units are in use, a node which could progress to READY state is permitted to "choose" not to, and to wait for other nodes to send ready signals. If such nodes are mis-configured, resulting in more than one node making such a choice, a distributed deadlock is possible that can only be detected by a timeout.

11.4.9 Rewrite intermediate record

This procedure may be invoked when the TPPM is in the DECIDED (commit) state, has received a *commit indication,* but the *last commit confirm* has not been received. These conditions imply that a log-ready record exists when this procedure is invoked.

The procedure writes a log-commit record replacing a log-ready record. An implementation that invokes this procedure shall ensure that any failure to write the log-commit record leaves the log-ready record intact, and also that if a failure of any kind occurs during this procedure, the recovery procedures act on either the old log-ready record or the new log-commit record. If the implementation of log-record writing is such that this behaviour cannot be ensured, this internal procedure shall not be invoked:

– *write a log-commit record* identifying as neighbours to whom a *ready signal was sent* at least all such neighbours from whom *commit confirm* has not yet been received.

NOTE – It is recognized that the attempt to write a log-commit record in the above paragraph may fail (but the log-ready record will remain undamaged). If the attempt does fail, this procedure is effectively null.

If the log-commit record was written, and for the dialogue on which the *commit indication* was received:

a) reporting does not apply; or

b) it is the superior dialogue and the reporting status is known; or

c) one of AF-REPORT (dataRI) indication, AF-ABORT-AND-REPORT (dataRI) indication or AF-REPORT (commitRI) indication has been received,

then:

– invoke the "Confirming commitment" procedure (see 11.5.2);

– *close the PSAP* for the dialogue on which *commit indication* was received, if that dialogue *has not been detached*.

11.4.10 Lazy log forget

This procedure may be invoked by the TPPM at any time after the *forget the transaction* action has been made pending:

– *forget the transaction*;

– make *forgetting the transaction* no longer pending.

11.5 Common procedures

These procedures are referenced by the main or internal event procedures.

Each procedure begins with a table that indicates the procedures that invoke the given procedure.

11.5.1 Confirm and complete commitment

Invoking Procedure Name	Subclause
TP-DONE request	11.3.50
C-COMMIT confirm or AF-REPORT (commitRC) indication	11.3.51
AF-ABORT (user, commitRC) indication or AF-ABORT-AND-REPORT (commitRC) indication	11.3.52
AF-REPORT (dataRI) indication or AF-ABORT-AND-REPORT (dataRI) indication	11.3.59
C-RECOVER (done) confirm or AF-REPORT (recoverDoneRC) indication	11.3.72

If *lazy-log-forget is applicable* and according to a local decision:

– make *forgetting the transaction* pending.

NOTE – Postponing the *forget the transaction* (i.e. performing a lazy log forget) implies that the TPPM may, following a node crash, issue a C-RECOVER (Ready) request for a transaction branch after it has issued a *commit-confirm.* In response, the TPPM may receive a C-RECOVER (unknown) confirm or a C-RECOVER (commit) indication, depending on the state of the partner. The C-RECOVER (unknown) confirm implies that the transaction has rolled back. The local decision of performing a lazy log thus requires that the TPPM guarantees that the ACID properties of the transaction will not be violated in this case.

Otherwise:

– *forget the transaction.*

Always:

– invoke the "Confirming commitment" procedure (see 11.5.2);

– invoke the "Completing commitment" procedure (see 11.5.3).

11.5.2 Confirming commitment

Invoking Procedure Name	Subclause
Rewriting intermediate record	11.4.9
Confirm and complete commitment	11.5.1

If the dialogue on which the *commit indication* was received is the superior dialogue and *has not been detached,* take the first applicable actions of the following in the scope of the dialogue with the superior:

1) If a TP-U-ABORT request has been received, no AF-ABORT indication was received, the Unchained Transactions functional unit is selected, and *a report is to be sent*.

   NOTE 1 – An AF-ABORT (user) request cannot have been previously issued (in response to a TP-U-ABORT request after the ACTIVE state) because there was no opportunity to issue a request to the superior after the TPPM entered the READY state.

   – Issue an AF-ABORT-AND-REPORT (commitRC) request *carrying the reporting status*;

   – issue an SAF-DETACH-ASSOCIATION (free) request.

2) If a TP-U-ABORT request was received in the READY state, no AF-ABORT indication was received, this is an intermediate node, a C-COMMIT+C-BEGIN indication was received and *a report is to be sent*:

   – issue an AF-ABORT-AND-REPORT (commitRC) request *carrying the reporting status*;

   – issue a C-ROLLBACK request;

   – issue an SAF-DETACH-ASSOCIATION (rollback-confirmation-expected) request.

3) If a TP-U-ABORT request has been received, no AF-ABORT indication was received, the Unchained Transactions functional unit is selected, and *a report is not to be sent*.

   NOTE 2 – An AF-ABORT (user) request cannot have been previously issued (in response to a TP-U-ABORT request after the ACTIVE state) because there was no opportunity to issue a request to the superior after the TPPM entered the READY state.

   – Issue an AF-ABORT (user, commitRC) request;

   – issue an SAF-DETACH-ASSOCIATION (free) request.

   NOTE 3 – The case of TP-U-ABORT request to the superior with the Chained Transactions functional unit selected is handled as part of the rollback procedures since the only way a TP-U-ABORT request could be issued to the superior is if there was a failure causing a rollback of the next transaction. In this way, the AF-ABORT indication to the superior will be carried by the appropriate CCR service.

4) If a TP-U-ABORT request was received in the READY state, no AF-ABORT indication was received, this is an intermediate node, a C-COMMIT+C-BEGIN indication was received and *a report is not to be sent*:

   – issue an AF-ABORT (user, commitRC) request;

   – issue a C-ROLLBACK request;

   – issue an SAF-DETACH-ASSOCIATION (rollback-confirmation-expected) request.

5) If *a report is to be sent*:

   – issue an AF-REPORT (commitRC) request *carrying the reporting status*;

   – issue an SAF-DETACH-ASSOCIATION (free) request if an AF-DEFER (end-dialogue) indication was received.

6) If none of the above conditions was met:

   – issue a C-COMMIT response;

   – issue an SAF-DETACH-ASSOCIATION (free) request if an AF-DEFER (end-dialogue) indication or AF-ABORT (user, commitRI) indication was received.

If a *one-phase indication* was received on the superior dialogue and that dialogue *has not been detached,* take the first applicable actions of the following in the scope of the dialogue with the superior:

1) If a TP-U-ABORT request has been received, no AF-ABORT indication was received, the Unchained Transactions functional unit is selected, and *reporting applies on the superior dialogue*:

   – issue an AF-ABORT-AND-REPORT (dataRI) request *carrying the reporting status*;

   – issue an SAF-DETACH-ASSOCIATION (free) request.

2) If a TP-U-ABORT request has been received, no AF-ABORT indication was received, the Unchained Transactions functional unit is selected, and *reporting does not apply on the superior dialogue*:

   – issue an AF-ABORT (user, dataRI) request;

   – issue an SAF-DETACH-ASSOCIATION (free) request.

3) If *reporting applies on the superior dialogue*:

   – issue an AF-REPORT (dataRI) request *carrying the reporting status*.

4) If none of the above apply:

   – continue.

If a *one-phase indication* was received on the superior dialogue and that dialogue *has been detached*:

– continue.

If the dialogue on which the *commit indication* was received is the dialogue to a subordinate and *has not been detached*:

a) If a TP-U-ABORT request has been received and no AF-ABORT (user, dataRI) indication or AF-ABORT-AND-REPORT indication was received:

   – issue an AF-ABORT (user, commitRC) request;

   – issue an SAF-DETACH-ASSOCIATION (free) request.

b) If a TP-U-ABORT request has been received and an AF-ABORT (user, dataRI) indication or AF-ABORT-AND-REPORT indication was received:

   – issue a C-COMMIT response;

   – issue an SAF-DETACH-ASSOCIATION (free) request.

c) If no TP-U-ABORT request has been received:

   – issue a C-COMMIT response;

   – issue an SAF-DETACH-ASSOCIATION (free) request if a TP-DEFERRED-END-DIALOGUE request or an AF-ABORT (user, commitRI) indication or AF-ABORT-AND-REPORT indication was received.

If the dialogue supporting the branch on which the *commit indication* was received *has been detached* and was the dialogue to the superior, and:

1) A channel with the superior is *attached*:

   – Issue a C-RECOVER (done) response if no *log-damage record exists* or if *heuristic reporting does not apply* to the branch.

   – Issue an AF-REPORT (recoverDoneRC) request if a log damage record exists and *heuristic reporting applies* to the branch. When issued the AF-REPORT (recoverDoneRC) request carries the Heuristic-Report parameter set to the current value of the log-damage record.

   – Issue a CAF-DETACH (free) request.

2) No channel with the superior is *attached*:

   – continue.

   NOTE 4 – In this case, when the superior subsequently establishes a channel and the CPM receives the C-RECOVER (commit) or AF-RECOVER (commit) indication, it will issue a C-RECOVER (done) response.

If the dialogue supporting the branch on which the *commit indication* was received *has been detached* and was a dialogue to a subordinate:

1) A channel with that subordinate is *attached*:

   – issue a C-RECOVER (done) response;

   – issue a CAF-DETACH (free) request.

2) No channel with that subordinate is *attached*:

   – continue.

11.5.3 Completing commitment

Invoking Procedure Name	Subclause
C-BEGIN indication or AF-BEGIN-TRANSACTION indication	11.3.37
Confirm and complete commitment	11.5.1
Making commitment decision	11.5.12
Making one-phase commitment decision	11.5.13

Always:

– issue a TP-COMMIT-COMPLETE indication.

Additionally:

a) If no TP-ROLLBACK indication is pending and:

1) any *dialogue is chaining*:

– enter the ACTIVE state;

2) no *dialogue is chaining*:

– cease to be part to the transaction;

3) the TPPM is an intermediate and the *dialogue* with the superior is *not chaining*:

– become a root node;

4) the TPPM is an intermediate and there are no *chaining subordinate dialogues*:

– become a leaf node;

5) for each *chaining subordinate dialogue* on which a C-NOCHANGE indication was received:

– issue a C-BEGIN request with the value of the:

- Atomic Action Identifier parameter set the first appropriate value of the following:

i) if the superior *dialogue is chaining*, the value of the atomic action identifier received on the most recent C-COMMIT+C-BEGIN indication;

ii) if any C-COMMIT+C-BEGIN request was issued, the value of the atomic action identifier on that request;

iii) a newly created value – This shall be the same value for all dialogues on which C-BEGIN request is issued;

- Branch Identifier parameter set to a value that uniquely identifies the transaction branch within the scope of the atomic action identifier.

b) If a TP-ROLLBACK indication is pending:

– issue a TP-ROLLBACK indication with the Diagnostic parameter set to the value "rollback-was-pending" if the Completion Diagnostics functional unit is selected on any dialogue;

– make the TP-ROLLBACK indication no longer pending;

– issue a C-CANCEL request on the superior dialogue, if that *dialogue has not been detached,* and on it no TP-U-ABORT request or C-CANCEL indication has been received, the cancel functional unit is selected and subject to a local decision;

– enter the DECIDED (rollback) state.

For each dialogue that has *closed the PSAP* and which *has not been detached*:

– *open the PSAP*.

For each dialogue on which AF-DEFER (end-dialogue) request or indication was issued and which *has not been detached*:

– issue an SAF-DETACH-ASSOCIATION (free) request.

11.5.4 Reporting on the commit-coordinator:root path

Invoking Procedure Name	Subclause
TP-DONE request	11.3.50
C-COMMIT confirm or AF-REPORT (commitRC) indication	11.3.51
AF-REPORT (dataRI) indication or AF-ABORT-AND-REPORT (dataRI) indication	11.3.59
C-RECOVER (done) confirm or AF-REPORT (recoverDoneRC) indication	11.3.72

If the superior dialogue *has not been detached* and a TP-U-ABORT request has been received for the superior dialogue:

– issue an AF-ABORT-AND-REPORT (dataRI) request *carrying the reporting status*;

– issue an SAF-DETACH-ASSOCIATION (free) request if the TPPM is in the DECIDED (commit-one-phase) state.

If the superior dialogue *has not been detached* and no TP-U-ABORT request has been received for the superior dialogue:

– issue an AF-REPORT (dataRI) request *carrying the reporting status.*

If the superior dialogue *has been detached* and no *one-phase indication* was received on that dialogue:

– issue a CAF-DETACH (clean-up) request on the channel to the superior, if one is *attached*;

– issue a CAF-PLEASE request with the AE-Title of the CAF-PLEASE request set to the value of the AE-Title taken from the branch identifier for the superior, contained in the log-commit record.

11.5.5 Completing ONE-PHASE and READ-ONLY

Invoking Procedure Name	Subclause
Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm on a dialogue	11.3.21
C-BEGIN indication or AF-BEGIN-TRANSACTION indication	11.3.37
TP-DONE request	11.3.50
AF-REPORT (dataRI) indication or AF-ABORT-AND-REPORT (dataRI) indication	11.3.59

For any dialogue on which an AF-DEFER (end-dialogue) request was issued or an AF-DEFER (end-dialogue) indication was received, no AF-EARLY-EXIT request was issued, no AF-EARLY-EXIT indication was received and which *has not been detached*:

– issue an SAF-DETACH-ASSOCIATION (free) request.

If the *dialogue* with the superior *is not chaining*:

a) If the dialogue with the superior *has not been detached*:

   – *open the PSAP*, if it is closed.

b) If any subordinate *dialogue is chaining* and no AF-NOCHANGE request was issued:

   – become a root node;

   – issue a C-BEGIN request to each *chaining subordinate dialogue* with the value of the:

      1) Atomic Action Identifier parameter set to a newly created value (the same value for all dialogues);

      2) Branch Identifier parameter set to a value that uniquely identifies the transaction branch within the scope of the atomic action identifier;

- enter the ACTIVE state.

  NOTE 1 – This situation occurs only when the TPPM was in the READ-ONLY state and the previously chaining superior dialogue (on which C-NOCHANGE request was issued) has been detached.

c) If there is exactly one subordinate *dialogue which is chaining* and on that dialogue an AF-NOCHANGE request was issued:

   - issue a C-BEGIN request to the *chaining subordinate dialogue* with the values of the Atomic Action Identifier parameter and the Branch Identifier parameters set to the values of the corresponding parameters on the previous AF-NOCHANGE request;

   - enter the ACTIVE state.

   NOTE 2 – The Chained and Static-one-phase functional units are selected on the subordinate dialogue. The TPPM is already the root.

d) If no subordinate *dialogue is chaining*:

   - cease to be part of the transaction.

If the superior *dialogue* is *chaining*:

a) Always:

   - *open the PSAP* with the superior, if closed.

b) If any subordinate *dialogue is chaining*:

   - issue a C-BEGIN request to each *chaining subordinate dialogue* with the value of the:

     1) Atomic Action Identifier parameter set to the value of the atomic action identifier received on the most recent C-BEGIN indication;

     2) Branch Identifier parameter set to a value that uniquely identifies the transaction branch within the scope of the atomic action identifier;

   - enter the ACTIVE state.

c) If no subordinate *dialogue is chaining*:

   - become a leaf node;

   - enter the ACTIVE state.

If a TP-COMMIT indication was issued:

- issue a TP-COMMIT-COMPLETE indication.

If a TP-UNKNOWN indication was issued:

- issue a TP-UNKNOWN-COMPLETE indication.

11.5.6 Entering READY state

Invoking Procedure Name	Subclause
TP-COMMIT request	11.3.45
AF-PREPARE indication	11.3.46
C-READY indication	11.3.47
TP-ONE-PHASE request	11.3.60
TP-READ-ONLY request	11.3.61
AF-NOCHANGE indication or C-NOCHANGE indication	11.3.62

NOTE 1 – This procedure is entered when there is one dialogue on which no C-READY indication or *ready-substitute indication* has been received. Normally, after writing a log-ready record, a C-READY request will be issued on that dialogue by this procedure.

If the TPPM is able to set the *TPPM bound data* in the ready-to-commit state:

- *Write a log-ready record* atomically setting the *TPPM bound data* to the ready-to-commit state, if they are not already set to the ready-to-commit state.

  NOTE 2 – If the TPPM had previously commenced setting the *TPPM bound data* to the ready-to-commit state, this completes that action.

– *Write a log-ready record* if the *TPPM bound data* are already set to the ready-to-commit state.

NOTE 3 – An implementation may choose to set the *TPPM bound data* to the ready-to-commit state atomically with writing the log-ready record.

NOTE 4 – If the log-ready record cannot be written, a rollback is triggered. This implies that the action of writing the log-ready record in the above two paragraphs may fail. That is why there is a check for the success or failure of writing the log-ready record in the following paragraphs.

If the log-ready record was written:

a) Always:

   – enter the READY state;

   – invoke the "First request/response" procedure (see 11.5.8), if an AF-BEGIN-DIALOGUE response has not been issued.

b) If C-READY indication or C-NOCHANGE indication has been received on all subordinate dialogues:

   – Issue a P-TOKEN-GIVE (sync-minor) request, if the *token* is owned by the node on the dialogue with the superior.

   NOTE 5 – If the *token* was moved by the User-ASE to the subordinate, issuance of a P-TOKEN-GIVE (sync-minor) request is necessary to allow the superior to issue a *commit request*, if required.

   – Issue a C-READY request on the dialogue with the superior.

c) If this is the root node or a C-READY indication or C-NOCHANGE indication has been received from the superior, then, on the subordinate dialogue on which no C-READY indication or *ready-substitute indication* has been received:

   – issue a P-TOKEN-GIVE (sync-minor) request, if the *token* is owned;

   – issue a C-READY request.

If the TPPM is unable to set the *TPPM bound data* in the ready-to-commit state or unable to write the log-ready record:

– invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "local-rollback".

11.5.7 Fail an outstanding CAF-PLEASE request

Invoking Procedure Name	**Subclause**
AF-BEGIN-DIALOGUE confirm (CPM)	11.3.8
AF-END-DIALOGUE indication (CPM)	11.3.13
Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm (CPM)	11.3.23
C-RECOVER (ready) indication or AF-RECOVER (ready) indication (CPM)	11.3.67
C-RECOVER (commit) indication or AF-RECOVER (commit) indication or AF-REPORT (recoverCommitRI) indication (CPM)	11.3.71

If a TPPM with a branch corresponding to the "Atomic Action Identifier", "Branch Identifier", and "Superior" parameters specified of the outstanding CAF-PLEASE request is found:

– issue a CAF-FAIL indication to the found TPPM,

otherwise:

– continue.

11.5.8 First request/response

Invoking Procedure Name	Subclause
TP-BEGIN-DIALOGUE response	11.3.3
TP-END-DIALOGUE request	11.3.11
TP-END-DIALOGUE response	11.3.14
TP-U-ERROR request	11.3.16
TP-U-ABORT request	11.3.19
TP-GRANT-CONTROL request	11.3.24
TP-REQUEST-CONTROL request	11.3.26
TP-HANDSHAKE request	11.3.28
TP-HANDSHAKE response	11.3.30
TP-HANDSHAKE-AND-GRANT-CONTROL request	11.3.32
TP-HANDSHAKE-AND-GRANT-CONTROL response	11.3.34
TP-DATA request	11.3.39
Entering READY state	11.5.6
Entering ONE-PHASE or READ-ONLY state	11.5.20

NOTE – This procedure is not used for the DECIDED (rollback) state.

- Issue an AF-BEGIN-DIALOGUE (accepted, dataRI) response;
- issue a C-BEGIN response if a C-BEGIN indication was received;
- issue as many AF-U-ERROR responses as the number of AF-U-ERROR indications that have been received, if the Shared Control functional unit is selected.

11.5.9 Initiating a transaction branch

Invoking Procedure Name	Subclause
TP-BEGIN-DIALOGUE request	11.3.1
TP-BEGIN-TRANSACTION request	11.3.36

If this procedure was invoked by a TP-BEGIN-TRANSACTION request on which the Check-ready-directions parameter was present and had the value "false":

- issue an AF-BEGIN-TRANSACTION request with the parameter values defined below,

otherwise:

- issue a C-BEGIN request with the parameter values defined below.

In either case with the value of the:

a) Atomic Action Identifier parameter set to:

   1) a newly created value, if the TPPM is not in the ACTIVE state;

   2) the value of the atomic action identifier of the most recent C-BEGIN indication or AF-BEGIN-TRANSACTION indication, if this is an intermediate or leaf node in the ACTIVE state; or

   3) the value of the atomic action identifier of the transaction, if this is a root node in the ACTIVE state.

b) Branch Identifier parameter set to a value that uniquely identifies the transaction branch within the scope of the atomic action identifier.

If the TPPM is a leaf node in the ACTIVE state:

- become an intermediate node.

If the TPPM is not in the ACTIVE state:

– become a root node;

– enter the ACTIVE state.

11.5.10 Initiating rollback at TPPM

Invoking Procedure Name	Subclause
AF-BEGIN-DIALOGUE (rejected, dataRI) confirm on a Dialogue	11.3.5
AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) confirm	11.3.6
SAF-ASSOCIATION-LOST indication	11.3.9
AF-END-DIALOGUE indication	11.3.12
AF-U-ERROR indication	11.3.17
TP-U-ABORT request	11.3.19
AF-ABORT (user, dataRI) indication	11.3.20
Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm on a dialogue	11.3.21
AF-HANDSHAKE indication	11.3.29
U-ASE indication	11.3.40
AF-DEFER indication	11.3.43
AF-PREPARE indication	11.3.46
C-READY indication	11.3.47
TP-ROLLBACK request	11.3.53
C-ROLLBACK indication or AF-REPORT (rollbackRI) indication	11.3.54
C-CANCEL indication	11.3.55
AF-ABORT (user/provider, rollbackRI) indication or AF-ABORT-AND-REPORT (rollbackRI) indication	11.3.56
C-ROLLBACK confirm or AF-REPORT (rollbackRC) indication	11.3.57
AF-ABORT (user/provider, rollbackRC) indication or AF-ABORT-AND-REPORT (rollbackRC) indication	11.3.58
AF-NOCHANGE indication or C-NOCHANGE indication	11.3.62
TP-EARLY-EXIT request	11.3.63
AF-EARLY-EXIT indication	11.3.64
CAF-RECOVER (ready) indication	11.3.66
C-RECOVER (unknown) confirm	11.3.73
TPPM-initiated rollback	11.4.8
Entering READY state	11.5.6
Making commitment decision	11.5.12

Invocations of this procedure that will result in the issue of a TP-ROLLBACK indication specify a value for "diagnostic-value". If the Completion Diagnostics functional unit is selected on any coordinated dialogue, the Diagnostic parameter on any TP-ROLLBACK indication issued by this procedure is set to the specified "diagnostic-value". If the "diagnostic-value" is "local-rollback", the Severity parameter on the TP-ROLLBACK indication is set to a locally-determined value. For any other diagnostic-value, and if the Completion Diagnostics functional unit is not selected on any dialogue, the Severity parameter on any TP-ROLLBACK indication shall be absent.

NOTE 1 – A *rollback request* or *rollback response* is not sent to the superior until a TP-DONE request has been issued by the TPSUI and a rollback indication or confirm has been received from each subordinate. This allows for the propagation of any heuristic condition and completion reporting information to the root of the transaction tree and for abort opportunities. Sending the *rollback request* or *rollback response* to the superior is done in the "Reporting rollback to superior" procedure (see 11.5.16). A C-CANCEL request can be sent to the superior as soon as the rollback result is known, if the cancel functional unit is selected on the dialogue with the superior and the *rollback request* is not sent as part of the same action sequence.

If the TPPM is in the ACTIVE or ONE-PHASE or READ-ONLY or EARLY-EXIT states:

– Enter the DECIDED (rollback) state.

– Issue a TP-ROLLBACK indication, if no rollback initiating request has been received nor rollback initiating indication has been issued (by the calling procedure).

NOTE 2 – The terms "rollback initiating request" and "rollback initiating indication" are defined in the TP-Service, ITU-T Rec. X.861 | ISO/IEC 10026-2.

– Set the *TPPM bound data*, if any, to the initial state.

– Issue a C-ROLLBACK response if this procedure was invoked as a result of receiving a rollback indication from a subordinate.

NOTE 3 – Thus, if an AF-EARLY-EXIT indication was received, but caused a rollback, the reply is a C-ROLLBACK response.

– Issue a C-ROLLBACK request to each subordinate except any to which a *rollback request* was issued or from which a *rollback indication* has been received or whose *dialogue has been detached*.

NOTE 4 – Unless an AF-EARLY-EXIT indication has been received, there can be at most one subordinate dialogue to which C-ROLLBACK request is not issued. The case of the *rollback request* being issued would occur only when the rollback is issued in the same action sequence by the calling procedure.

– Issue a C-CANCEL request on the superior dialogue, if the dialogue is still *attached*, no *rollback indication* has been received from the superior, the cancel functional unit is selected on the dialogue and subject to a local decision.

If the TPPM is in the READY state:

– Enter the DECIDED (rollback) state.

– Issue a TP-ROLLBACK indication, if no rollback initiating indication has been issued (by the calling procedure).

NOTE 5 – The term "rollback initiating indication" is defined in the TP Service, ITU-T Rec. X.861 | ISO/IEC 10026-2.

– Set the *TPPM bound data*, if any, to the initial state, unless a heuristic decision has been taken or the bound data have already been set to the initial state by a previous invocation of this procedure.

NOTE 6 – A node crash occurring during the execution of this procedure before the log-ready record is removed will lead to another invocation of this procedure as part of the recovery protocol. If necessary, some information must be stored in secure storage to ensure that another invocation of this procedure does not repeat setting the data to the initial state.

– Issue a C-ROLLBACK response if this is a *rollback indication* from a subordinate.

– Issue a C-ROLLBACK request to each subordinate for which no TP-U-ABORT request was received, from which no *rollback indication* has been received and whose *dialogue has not been detached*.

– Issue a C-CANCEL request on the superior dialogue, if the dialogue is still *attached,* the Cancel functional unit is selected on the dialogue and subject to a local decision.

NOTE 7 – Since the TPPM is in the READY state, this can only occur after a C-READY indication has been received from the superior.

– Issue an AF-ABORT (user, rollbackRI) request to each subordinate for which a TP-U-ABORT request was received and whose *dialogue has not been detached*.

– Issue a C-RECOVER (unknown) response to each subordinate from which a CAF-RECOVER (ready) indication was received and to which no C-RECOVER (retry-later) response was issued.

- Issue a CAF-DETACH (free) request to each subordinate from which a CAF-RECOVER (ready) indication was received and to which no C-RECOVER (retry-later) response was issued and whose channel is still *attached*.

- *Write a log-damage record* with the value "heuristic-hazard" if one does not exist and a subordinate *dialogue* on which *heuristic reporting applied has been detached*.

- *Write a log-damage record* with a value of "heuristic-mix" if the bound data are set to a state other than the initial state as a result of a previously taken heuristic decision by the TPPM.

- *Forget the transaction*.

11.5.11 Initiating transaction after rollback

Invoking Procedure Name	Subclause
AF-BEGIN-DIALOGUE (accepted) confirm on a Dialogue	11.3.4
AF-BEGIN-DIALOGUE (rejected, dataRI) confirm on a Dialogue	11.3.5
AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) confirm	11.3.6
AF-BEGIN-DIALOGUE (rejected(user), rollbackRC) confirm	11.3.7
SAF-ASSOCIATION-LOST indication	11.3.9
AF-END-DIALOGUE indication	11.3.12
AF-ABORT (user, dataRI) indication	11.3.20
Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm on a dialogue	11.3.21
C-BEGIN indication or AF-BEGIN-TRANSACTION indication	11.3.37
TP-DONE request	11.3.50
C-ROLLBACK indication or AF-REPORT (rollbackRI) indication	11.3.54
AF-ABORT (user/provider, rollbackRI) indication or AF-ABORT-AND-REPORT (rollbackRI) indication	11.3.56
C-ROLLBACK confirm or AF-REPORT (rollbackRC) indication	11.3.57
AF-ABORT (user/provider, rollbackRC) indication or AF-ABORT-AND-REPORT (rollbackRC) indication	11.3.58
AF-EARLY-EXIT indication	11.3.64

- *Open the PSAP* for all subordinate dialogues for which the Unchained Transactions functional unit is selected;

If the *dialogue* with the superior *is not chaining*:

a) If the dialogue with the superior *has not been detached*:

  - *open the PSAP*, if it is closed.

b) If any subordinate *dialogue is chaining*:

  - become a root node;

  - issue a C-BEGIN request to each *chaining subordinate dialogue* with the value of the:

    1) Atomic Action Identifier parameter set to a newly created value;

    2) Branch Identifier parameter set to a value that uniquely identifies the transaction branch within the scope of the atomic action identifier;

  - enter the ACTIVE state.

c) If no subordinate *dialogue is chaining*:

– cease to be part of the transaction.

If the superior *dialogue is chaining*:

a) Always:

– *open the PSAP* with the superior, if closed.

b) If any subordinate *dialogue is chaining*:

– issue a C-BEGIN request to each *chaining subordinate dialogue* with the value of the:

1) Atomic Action Identifier parameter set to the value of the atomic action identifier received on the most recent C-BEGIN indication;

2) Branch Identifier parameter set to a value that uniquely identifies the transaction branch within the scope of the atomic action identifier;

– enter the ACTIVE state.

c) If no subordinate *dialogue is chaining*:

– become a leaf node;

– enter the ACTIVE state.

Always:

– Issue a TP-ROLLBACK-COMPLETE indication.

11.5.12 Making commitment decision

Invoking Procedure Name	Subclause
TP-COMMIT request	11.3.45
C-READY indication	11.3.47
TP-ONE-PHASE request	11.3.60
TP-READ-ONLY request	11.3.61
AF-NOCHANGE indication or C-NOCHANGE indication	11.3.62

If the TPPM is in the READY state:

– remove the log-ready record.

If the TPPM is able to set the *TPPM bound data* in the final state:

– *Write a log-commit record*.

NOTE 1 – If the log-commit record cannot be written, a rollback is triggered. This implies that the action of writing a log-commit record in the above paragraph may fail. That is why there is a check for the success or failure of writing the log-commit record in the following paragraphs.

NOTE 2 – The TPPM will only be in the READY state after a collision of ready signals. The replacement of the log-ready record by a log-commit record may be performed atomically, or the log-ready record can even be left, provided that regardless of when failures occur, the recovery mechanisms of the "Restart after node crash (CPM)" procedure do not attempt to act on both records.

– Begin setting the *TPPM bound data* to the final state, unless a heuristic decision has been taken. The *TPPM bound data* shall eventually be set to the final state; when this occurs is a local matter.

If the TPPM is unable to set the *TPPM bound data* in the final state or *write the log-commit record*:

– invoke the "Initiating rollback at TPPM" procedure (see 11.5.10) with a diagnostic-value of "local-rollback".

If the log-commit record was written:

– issue a TP-COMMIT indication;

– enter the DECIDED (commit) state;

– invoke the "Sending commit order" procedure (see 11.5.18).

If the *last commit confirm has been received*:

– invoke the "Completing commitment" procedure (see 11.5.3).

NOTE 3 – This can only occur if no C-READY indications were received and either TP-ONE-PHASE request or TP-READ-ONLY request was received, but there was TPPM bound data.

11.5.13 Making one-phase commitment decision

Invoking Procedure Name	**Subclause**
TP-ONE-PHASE request	11.3.60
TP-READ-ONLY request	11.3.61
AF-NOCHANGE indication or C-NOCHANGE indication	11.3.62

– Invoke the "Sending commit order" procedure (see 11.5.18);

– invoke the "Completing commitment" procedure (see 11.5.3).

11.5.14 Receiving commit order

Invoking Procedure Name	**Subclause**
C-COMMIT indication or C-COMMIT+C-BEGIN indication	11.3.48
AF-ABORT (user, commitRI) indication	11.3.49
CAF-RECOVER (commit) indication	11.3.69
C-RECOVER (commit) indication or AF-REPORT (recoverCommitRI) indication	11.3.70

Always:

– issue a TP-COMMIT indication.

If the process of setting the *TPPM bound data* to the final state has been commenced by a previous invocation of this procedure:

– continue setting the *TPPM bound data* to the final state.

If the process of setting the *TPPM bound data* to the final state has not been commenced by a previous invocation of this procedure:

– Begin setting the *TPPM bound data* to the final state, unless a heuristic decision has been taken. The *TPPM bound data* shall eventually be set to the final state; when this occurs is a local matter.

– *Write a log-damage record* with the value "heuristic-mix", if the node took a heuristic decision to set its *TPPM bound data* in a state other than the final state.

Always:

NOTE 1 – the TPPM of an intermediate or leaf node may or may not *write the log-commit record* at this time. Upon node crash recovery, ("Restart after node crash", see 11.4.3) if the log-commit record has not been written, the node will be in the READY state and will take the appropriate recovery actions.

– Enter the DECIDED (commit) state.

NOTE 2 – If the log-ready record is not replaced by a log-commit record, then with the start of the process to set the TPPM bound data to the final state, some information must be stored in secure storage to make it impossible for this procedure to start the process again during a subsequent invocation. If the log-ready record is not replaced by a log-commit record, then this procedure can be repeated multiple times.

11.5.15 Recording the heuristic condition

Invoking Procedure Name	Subclause
TP-DONE request	11.3.50
C-COMMIT confirm or AF-REPORT (commitRC) indication	11.3.51
AF-ABORT (user, commitRC) indication or AF-ABORT-AND-REPORT (commitRC) indication	11.3.52
C-ROLLBACK indication or AF-REPORT (rollbackRI) indication	11.3.54
AF-ABORT (user/provider, rollbackRI) indication or AF-ABORT-AND-REPORT (rollbackRI) indication	11.3.56
C-ROLLBACK confirm or AF-REPORT (rollbackRC) indication	11.3.57
AF-ABORT (user/provider, rollbackRC) indication or AF-ABORT-AND-REPORT (rollbackRC) indication	11.3.58
AF-REPORT (dataRI) indication or AF-ABORT-AND-REPORT (dataRI) indication	11.3.59
CAF-RECOVER (ready) indication	11.3.66
CAF-RECOVER (commit) indication	11.3.69
C-RECOVER (commit) indication or AF-REPORT (recoverCommitRI) indication	11.3.70
C-RECOVER (done) confirm or AF-REPORT (recoverDoneRC) indication	11.3.72

If the TPPM is in the DECIDED (commit-one-phase) state, any log-damage record "written" does not need to survive a node crash. In all other cases, the log-damage record is written to secure storage.

If the TPPM determines that heuristic damage can be compensated:

– continue.

If the Heuristic-report parameter is set to "heuristic-hazard":

– *write a log-damage record* with the value "heuristic-hazard" if it has not already been created.

If the Heuristic-report parameter is set to "heuristic-mix":

– *write log-damage record* with the value "heuristic-mix" if it has not already been created, or update it to "heuristic-mix" if its current value is "heuristic-hazard".

11.5.16 Reporting rollback to superior

Invoking Procedure Name	Subclause
AF-BEGIN-DIALOGUE (accepted) confirm on a Dialogue	11.3.4
AF-BEGIN-DIALOGUE (rejected, dataRI) confirm on a Dialogue	11.3.5
AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) confirm	11.3.6
AF-BEGIN-DIALOGUE (rejected(user), rollbackRC) confirm	11.3.7
SAF-ASSOCIATION-LOST indication	11.3.9
AF-END-DIALOGUE indication	11.3.12
AF-ABORT (user, dataRI) indication	11.3.20
Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm on a dialogue	11.3.21
TP-DONE request	11.3.50
C-ROLLBACK indication or AF-REPORT (rollbackRI) indication	11.3.54
AF-ABORT (user/provider, rollbackRI) indication or AF-ABORT-AND-REPORT (rollbackRI) indication	11.3.56
C-ROLLBACK confirm or AF-REPORT (rollbackRC) indication	11.3.57
AF-ABORT (user/provider, rollbackRC) indication or AF-ABORT-AND-REPORT (rollbackRC) indication	11.3.58
AF-EARLY-EXIT indication	11.3.64

If a log-ready record exists:

– *forget the transaction*.

If a *rollback indication* was received from the superior:

a) Take the first applicable set of actions on the dialogue to the superior:

1) If a TP-U-ABORT request has been received for the superior and no AF-ABORT indication was received from the superior and *a report is to be sent*:

– issue an AF-ABORT-AND-REPORT (rollbackRC) request *carrying the reporting status*;

– issue an SAF-DETACH-ASSOCIATION (free) request.

2) If a TP-U-ABORT request has been received for the superior and no AF-ABORT indication was received from the superior:

– issue an AF-ABORT (user, rollbackRC) request;

– issue an SAF-DETACH-ASSOCIATION (free) request.

3) If a *report is to be sent*:

– issue an AF-REPORT (rollbackRC) request *carrying the reporting status*;

– issue an SAF-DETACH-ASSOCIATION (free) request if an AF-ABORT indication was received.

4) If no AF-ABORT indication was received from the superior and the Confirmation parameter of the AF-BEGIN-DIALOGUE indication was set to "always" and no AF-BEGIN-DIALOGUE response has been issued:

NOTE – The service definition "dialogue establishment indication outstanding" does not apply here.

– issue an AF-BEGIN-DIALOGUE (accepted, rollbackRC) response.

5) If none of the above conditions could be met:

   – issue a C-ROLLBACK response;

   – issue an SAF-DETACH-ASSOCIATION (free) request if an AF-ABORT indication was received.

b) If the Unchained Transactions functional unit is selected on this dialogue and *a TP-DONE request is owed*:

   – *close the PSAP*.

If no *rollback indication* was received from the superior, issue the first applicable request of the following to the superior:

a) If a TP-U-ABORT request has been received for the superior and a *report is to be sent*:

   – issue an AF-ABORT-AND-REPORT (rollbackRI) request *carrying the reporting status*.

b) If a TP-U-ABORT request has been received for the superior:

   – issue an AF-ABORT (user, rollbackRI) request.

c) If *a report is to be sent*:

   – issue an AF-REPORT (rollbackRI) request *carrying the reporting status*.

d) If none of the above conditions could be met:

   – issue a C-ROLLBACK request.

11.5.17 Rollback next transaction

Invoking Procedure Name	Subclause
TP-U-ABORT request	11.3.19
AF-ABORT (user, dataRI) indication	11.3.20
Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm on a dialogue	11.3.21
CAF-RECOVER (ready) indication	11.3.66
CAF-RECOVER (commit) indication	11.3.69

NOTE 1 – This procedure is only invoked when the TPPM is in the DECIDED (commit) state.

If a TP-ROLLBACK indication is not pending:

– make a TP-ROLLBACK indication pending.

NOTE 2 – The pending TP-ROLLBACK indication is issued at the completion of this transaction in the "Completing commitment" procedure (see 11.5.3).

If the TPPM has no subordinates:

– continue.

For each subordinate *dialogue* that *has not been detached* that meets the following conditions:

a) a C-COMMIT+C-BEGIN request was issued and a *commit confirm* was received; and

b) no C-ROLLBACK request or AF-ABORT request was issued,

then:

a) If no TP-U-ABORT request was received:

   – issue a C-ROLLBACK request.

b) If a TP-U-ABORT request was received:

   – issue an AF-ABORT (user, rollbackRI) request;

   – issue an SAF-DETACH-ASSOCIATION (rollback-confirm-expected) request.

For each subordinate *dialogue* that *has not been detached* that meets the following conditions:

a) a C-COMMIT+C-BEGIN request was issued and no *commit confirm* was received; and

b) no C-ROLLBACK request or AF-ABORT request was issued; and

c) no TP-U-ABORT request was received; and

d) the cancel functional unit is selected; and

e) subject to a local decision,

then:

– issue a C-CANCEL request.

11.5.18 Sending commit order

Invoking Procedure Name	Subclause
C-COMMIT indication or C-COMMIT+C-BEGIN indication	11.3.48
AF-ABORT (user, commitRI) indication	11.3.49
C-NOCHANGE confirm or AF-ABORT (user, nochangeRC) indication	11.3.68
CAF-RECOVER (commit) indication	11.3.69
C-RECOVER (commit) indication or AF-REPORT (recoverCommitRI) indication	11.3.70
Making commitment decision	11.5.12

This action sequence can require C-COMMIT+C-BEGIN request or C-BEGIN request to be issued on one or more dialogues. For all such primitives issued on subordinate dialogues, the Atomic Action Identifier parameter is set to the same value, which shall be:

a) If this is a root node or the superior *dialogue is not chaining*:

   – a newly created value.

b) If the superior *dialogue is chaining*:

   – the value of the atomic action identifier of the most recent C-COMMIT+C-BEGIN indication or AF-NOCHANGE indication.

The Branch Identifier parameter in each primitive issued on a subordinate dialogue shall be set to a value that uniquely identifies the transaction branch within the scope of the Atomic Action Identifier.

NOTE 1 – If an AF-NOCHANGE indication was received on a *chaining* superior dialogue, the Atomic Action Identifier received on the AF-NOCHANGE indication is used for the C-COMMIT+C-BEGIN requests and will later be received on a C-BEGIN on the superior dialogue.

For each subordinate dialogue on which a *ready signal was received* and no *commit-indication was received*:

a) If the *dialogue is chaining*, and the *token* is owned on the dialogue:

   – issue a C-COMMIT+C-BEGIN request.

b) If the *dialogue is not chaining* and:

   1) if the *dialogue has not been detached* and:

      i) if a TP-U-ABORT request has been received for this subordinate:

         – issue an AF-ABORT (user, commitRI) request to the subordinate;

      ii) if a TP-U-ABORT request has not been received for this subordinate:

         – issue a C-COMMIT request to the subordinate;

   2) if the *dialogue has been detached* and:

      i) if a channel is not *attached*:

         – issue a CAF-PLEASE request with the AE-Title of the CAF-PLEASE request set to the value of the AE-Title taken from the branch identifier for the subordinate, contained in the log-commit record;

      ii) if a channel is *attached*:

         – issue a C-RECOVER (commit) request.

      NOTE 2 – Since the channel is *attached*, a CAF-RECOVER (ready) indication will have been received and CCR permits the issuance of a C-RECOVER (commit) request without the *token* when following a C-RECOVER (ready) indication.

For each subordinate dialogue on which C-NOCHANGE (result-requested) indication was received and which *has not been detached*:

a) If a TP-U-ABORT request has been received for this subordinate:

   – issue an AF-ABORT (user, nochangeRC) request with the Outcome parameter set to "commit".

b) If no TP-U-ABORT request has been received for this subordinate:

   – issue a C-NOCHANGE (commit) response.

For each subordinate dialogue on which a C-NOCHANGE (not-required) indication was received, no C-NOCHANGE response has been issued and which *has not been detached*:

a) If the *dialogue is chaining*:

   – continue.

b) If the *dialogue is not chaining*:

   i) if a TP-U-ABORT request has been received for this subordinate:

      – issue an AF-ABORT (user, nochangeRC) request with the Outcome parameter set to "not-determined";

      – issue an SAF-DETACH-ASSOCIATION (free) request;

   ii) if no TP-U-ABORT request has been received for this subordinate:

      – issue a C-NOCHANGE (not-determined) response;

      – issue an SAF-DETACH-ASSOCIATION (free) request if an AF-DEFER (end-dialogue) request was issued.

   NOTE 3 – The implementation may choose to delay the effect of the C-NOCHANGE response (i.e. the transmission of C-NOCHANGE-RC PDU) until some other outbound event occurs on the dialogue. Such a delay is not explicitly modelled in this Protocol Specification.

If a C-READY indication was received on the superior dialogue and no *commit-indication was received* on that dialogue:

NOTE 4 – This can only occur with dynamic commit which cannot be combined with chained transactions, so there is no possibility of the dialogue chaining.

a) If the superior *dialogue has not been detached*:

   i) if a TP-U-ABORT request has been received for this dialogue:

      – issue an AF-ABORT (user, commitRI) request to the superior;

ii) if a TP-U-ABORT request has not been received for this dialogue:

- issue a C-COMMIT request and a P-TOKEN-GIVE request to the superior.

b) If the superior dialogue *has been detached*:

i) if a channel is not *attached*:

- issue a CAF-PLEASE request with the AE-Title of the CAF-PLEASE request set to the value of the AE-Title taken from the branch identifier for the superior, contained in the log-commit record;

ii) if a channel is *attached*:

- issue a C-RECOVER (commit) request if no recovery-context-handle is present in the log-ready record;

NOTE 5 – Since the channel is *attached*, a CAF-RECOVER (ready) indication will have been received and CCR permits the issuance of a C-RECOVER (commit) request without the *token* when following a C-RECOVER (ready) indication. If the *token* is owned and the channel is two-way-recovery, it will be returned after the C-RECOVER (commit) request, as specified below.

- issue an AF-TOKEN-GIVE (two-way-recovery) request if the *token* is owned on the channel and this is a two-way-recovery channel.

If a *one-phase indication* was received on the superior dialogue, no C-NOCHANGE request was issued on that dialogue and that dialogue *has not been detached*:

a) If a TP-U-ABORT request has been received for this superior dialogue and *reporting does not apply on the dialogue*:

- issue an AF-ABORT (user, nochangeRC) request with the Outcome parameter set to "commit";

- issue an SAF-DETACH-ASSOCIATION (begin-fear) request.

b) If no TP-U-ABORT request has been received for this superior dialogue or *reporting does apply on the dialogue*:

- issue a C-NOCHANGE (commit) response;

- issue an SAF-DETACH-ASSOCIATION (free) request, if an AF-DEFER (end-dialogue) indication was received.

NOTE 6 – If a TP-U-ABORT request has been received, but reporting applies, the abort will be sent with the report.

11.5.19 Sending not-determined result from a ONE-PHASE or READ-ONLY node

Invoking Procedure Name	Subclause
Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm on a dialogue	11.3.21
AF-EARLY-EXIT confirm	11.3.65
C-NOCHANGE confirm or AF-ABORT (user, nochangeRC) indication	11.3.68

For each subordinate dialogue on which a C-NOCHANGE indication was received, no C-NOCHANGE response has been issued and which *has not been detached*:

a) If the *dialogue is chaining*:

- continue.

b) If the *dialogue is not chaining*:

i) if a TP-U-ABORT request has been received for this subordinate:

- issue an AF-ABORT (user, nochangeRC) request with the Outcome parameter set to "not-determined";
- issue an SAF-DETACH-ASSOCIATION (free) request;

ii) if no TP-U-ABORT request has been received for this subordinate:

- issue a C-NOCHANGE (not-determined) response;
- issue an SAF-DETACH-ASSOCIATION (free) request if an AF-DEFER (end-dialogue) request was issued.

If, on the superior dialogue, a C-NOCHANGE indication was received, no C-NOCHANGE response has been issued and the dialogue *has not been detached*:

a) If the *dialogue is chaining*:

- issue a C-NOCHANGE (not-determined) response.

b) If the *dialogue is not chaining*:

i) if a TP-U-ABORT request has been received for this subordinate:

- issue an AF-ABORT (user, nochangeRC) request with the Outcome parameter set to "not-determined";
- issue an SAF-DETACH-ASSOCIATION (free) request;

ii) if no TP-U-ABORT request has been received for this subordinate:

- issue a C-NOCHANGE (not-determined) response;
- issue an SAF-DETACH-ASSOCIATION (free) request if an AF-DEFER (end-dialogue) indication was received.

11.5.20 Entering ONE-PHASE or READ-ONLY state

Invoking Procedure Name	Subclause
AF-PREPARE indication	11.3.46
TP-ONE-PHASE request	11.3.60
TP-READ-ONLY request	11.3.61
AF-NOCHANGE indication or C-NOCHANGE indication	11.3.62

Always:

- enter the ONE-PHASE state if a TP-ONE-PHASE request was received or a C-NOCHANGE (result-requested) indication was received;
- enter the READ-ONLY state if a TP-READ-ONLY request was received and no C-NOCHANGE (result-requested) indication was received;
- invoke the "First request/response" procedure (see 11.5.8), if an AF-BEGIN-DIALOGUE response has not been issued.

If the one dialogue on which no *ready-substitute indication* has been received is a subordinate dialogue and:

a) The *dialogue is not chaining*:

   – issue a C-NOCHANGE request with the Confirmation parameter set to "result-requested".

b) If the *dialogue is chaining* and this is the root node:

   – issue an AF-NOCHANGE request with the:

      1) Atomic Action Identifier parameter set to a newly created value;

      2) Branch Identifier parameter set to a value that uniquely identifies the transaction branch (of the future transaction) within the scope of the atomic action identifier.

   NOTE – The restrictions on functional unit combinations, and on the establishment of static one-phase dialogues mean there cannot be a chaining subordinate dialogue for a non-root node in ONE-PHASE.

If the one dialogue on which no *ready-substitute indication* has been received is the superior dialogue, then on that dialogue:

– issue a C-NOCHANGE request with the Confirmation parameter set to the first applicable of the following cases:

   i) "result-requested", if a TP-ONE-PHASE request has been received or at least one C-NOCHANGE indication with the Confirmation parameter set to "result-requested" has been received;

   ii) "result-not-required", if a TP-READ-ONLY request was received and no C-NOCHANGE indication with the Confirmation parameter set to "result-requested" has been received.

11.5.21 User protocol error

Invoking Procedure Name	Subclause
AF-HANDSHAKE confirm	11.3.31
AF-HANDSHAKE-AND-GRANT-CONTROL indication	11.3.33
U-ASE indication	11.3.40
AF-DEFER indication	11.3.43

NOTE – This procedure is only invoked when the TPSUIs have violated application semantics and issued primitives in the wrong sequence, and the TPSP is unable to detect this until after a C-READY request or *ready-substitute request* has been issued by the TPPM that does detect the error. This may allow a commit decision to be taken by another TPPM. The "user-protocol-error" diagnostic thus indicates that the application is corrupt. This procedure aborts the dialogue and, if no commit decision has been taken, may cause the transaction to rollback, which will be determined locally via the recovery procedures. Implementations may allow management intervention, rather than complete this procedure.

Always:

– issue an AF-ABORT (provider, abortRI) request with the Diagnostic parameter set to "user-protocol-error";

– issue a TP-P-ABORT indication with the Rollback parameter set to "false" and the Diagnostic parameter set to "user-protocol-error".

If a *ready signal has been sent* on the dialogue:

– issue a CAF-PLEASE request with the AE-Title of the CAF-PLEASE request set to the value of the AE-Title taken from the branch identifier for the neighbour to which the *ready signal was sent* contained in the log-ready record.

If a TP-ONE-PHASE request or TP-READ-ONLY request was received:

– issue a TP-UNKNOWN indication.

12 Structure and encoding of TP APDUs

12.1 Abstract syntax of the TPASE APDUs

```
Transaction-Processing-APDUs

{joint-iso-itu-t transaction-processing(10) modules(1) apdus-abstract-syntax(1) version3(2)}

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- EXPORTS
-- all definitions --

IMPORTS
        APPLICATION-SERVICE-ELEMENT
            FROM    Remote-Operations-Notation-extension
            {joint-iso-ccitt remote-operation(4) notation-extension(2)}

-- object identifier assignments

id-as-tpase OBJECT IDENTIFIER ::=
        {joint-iso-itu-t transaction-processing(10) abstract-syntax(2) tp-apdus(1)}

            -- may be used to reference the abstract syntax of the
            -- transaction processing ASE APDUs

tpASE1 APPLICATION-SERVICE-ELEMENT ::=
        {joint-iso-itu-t transaction-processing(10) ase-id(0) tp-ase(1) version1(0)}

            -- may be used to identify the transaction processing ASE

-- top level APDU CHOICE

TPASE-APDU ::= CHOICE
{
tp-begin-dialogue-ri                   [1]   TP-BEGIN-DIALOGUE-RI,
tp-begin-dialogue-rc                   [2]   TP-BEGIN-DIALOGUE-RC,
tp-bid-ri                              [3]   TP-BID-RI,
tp-bid-rc                              [4]   TP-BID-RC,
tp-end-dialogue-ri                     [5]   TP-END-DIALOGUE-RI,
tp-end-dialogue-rc                     [6]   TP-END-DIALOGUE-RC,
tp-u-error-ri                          [7]   TP-U-ERROR-RI,
tp-u-error-rc                          [8]   TP-U-ERROR-RC,
tp-abort-ri                            [9]   TP-ABORT-RI,
tp-grant-control-ri                    [10]  TP-GRANT-CONTROL-RI,
tp-request-control-ri                  [11]  TP-REQUEST-CONTROL-RI,
tp-handshake-ri                        [12]  TP-HANDSHAKE-RI,
tp-handshake-rc                        [13]  TP-HANDSHAKE-RC,
tp-handshake-and-grant-control-ri      [14]  TP-HANDSHAKE-AND-GRANT-CONTROL-RI,
tp-handshake-and-grant-control-rc      [15]  TP-HANDSHAKE-AND-GRANT-CONTROL-RC,
tp-defer-ri                            [16]  TP-DEFER-RI,
tp-prepare-ri                          [17]  TP-PREPARE-RI,
tp-report-ri                           [18]  TP-REPORT-RI,
tp-token-give-ri                       [19]  TP-TOKEN-GIVE-RI,
tp-token-please-ri                     [20]  TP-TOKEN-PLEASE-RI,
tp-recover-ri                          [21]  TP-RECOVER-RI,
tp-initialize-ri                       [22]  TP-INITIALIZE-RI,
tp-initialize-rc                       [23]  TP-INITIALIZE-RC,
...,
tp-begin-transaction-ri                [24]  TP-BEGIN-TRANSACTION-RI,
tp-next-tid-ri                         [25]  TP-NEXT-TID-RI,
tp-abort-and-report-ri                 [26]  TP-ABORT-AND-REPORT-RI,
tp-solicit-dialogue-ri                 [27]  TP-SOLICIT-DIALOGUE-RI,
tp-solicit-dialogue-rc                 [28]  TP-SOLICIT-DIALOGUE-RC
}
```

```
-- individual APDU definitions

TP-BEGIN-DIALOGUE-RI ::= SEQUENCE
{       CHOICE
            {dialogue               [1]  SEQUENCE
                    {initiating-tpsu-title     [1]  TPSU-title OPTIONAL,
                     recipient-tpsu-title      [2]  TPSU-title OPTIONAL,
                     functional-units          [3]  FU-list DEFAULT
                                                        {shared-control,
                                                         commit-and-chained-transactions},
                -- dialogue is always selected.
                -- "recovery" shall not be selected.
                -- At most one of
                -- "commit-and-chained-transactions"
                -- "commit-and-unchained-transactions"
                -- "one-phase-commit-and-chained-transactions" and
                -- "one-phase-commit-and-unchained-transactions"
                -- shall be selected.
                 begin-transaction              [4]  BOOLEAN OPTIONAL,
                 confirmation                   [5]  ENUMERATED
                                                        {always       (1),
                                                         negative     (2)
                                                        }     DEFAULT  negative,
                 correlator                     [6]  Correlator,
                 last-partner-identifier        [7]  Correlator OPTIONAL,
                 ...,
                 superior-may-send-ready        [8]  BOOLEAN DEFAULT FALSE,
                 subordinate-may-send-ready     [9]  BOOLEAN DEFAULT TRUE,
                 check-ready-directions         [10] Check-ready-directions
                                                        DEFAULT TRUE,
                 recovery-context-handle        [11] Recovery-context-handle
                                                        OPTIONAL,
                 ...,
                 user-data                      [30] User-information OPTIONAL
                },

             channel                [2]  SEQUENCE
                {functional-units               [1]  FU-list DEFAULT {recovery},
                    -- Only Recovery shall be selected.
                    -- Default is Recovery.
                 correlator                     [2]  Correlator,
                 channel-utilization            [3]  ENUMERATED
                                                        {one-way-recovery  (1),
                                                         two-way-recovery  (2),
                                                         ...
                                                        } DEFAULT one-way-recovery,
                 last-partner-identifier        [4]  Correlator OPTIONAL,
                 ...
                }
            }
}

TP-BEGIN-DIALOGUE-RC ::= SEQUENCE
{       CHOICE
            {dialogue               [1]  SEQUENCE
                    {functional-units          [1]  FU-list OPTIONAL,
                        -- "recovery" shall not be selected.
                     result                    [2]  ENUMERATED
                                                        {accepted              (1),
                                                         rejected-provider     (2),
                                                         rejected-user         (3)
                                                        } DEFAULT  accepted,
                         diagnostic                [3]   ENUMERATED
                                     {recipient-tpsu-title-unknown                (1),
                                      tpsu-not-available-permanent                (2),
                                      tpsu-not-available-transient                (3),
                                      recipient-tpsu-title-required               (4),
                                      functional-unit-not-supported               (5),
                                      functional-unit-combination-not-supported   (6),
                                      association-reserved                        (7),
                                      no-reason-given                             (8),
                                      ...
                                     } OPTIONAL,
                        correlator                 [4]   Correlator,
                        ...,
                        recovery-context-handle    [5]   Recovery-context-handle
                                                                     OPTIONAL,
                        ...,
                        user-data                  [30]  User-information OPTIONAL
                             },
              channel                     [2]   SEQUENCE
                        {result                    [1]   ENUMERATED
                                                         {accepted             (1),
                                                          rejected-provider    (2)
                                                         } DEFAULT  accepted,
                               diagnostic          [2]   ENUMERATED,
                                     {functional-unit-not-supported               (1),
                                      association-reserved                        (2),
                                      tppm-recovery-not-available                 (3),
                                      two-way-recovery-not-supported              (4),
                                      no-reason-given                             (5),
                                      ...
                                     }  OPTIONAL,
                               correlator          [3]   Correlator,
                                      ...
                        }
                   }
}

TP-BID-RI  ::=  SEQUENCE
{                       ccr-token-requested        [1]   BOOLEAN  DEFAULT  FALSE,
                        last-partner-identifier    [2]   Correlator OPTIONAL,
                   ...
}

TP-BID-RC ::=  SEQUENCE
{                       result                     [1]   ENUMERATED
                                                         {accepted             (1),
                                                          rejected             (2)
                                                         } DEFAULT accepted,
                        ...
}

TP-END-DIALOGUE-RI  ::=  SEQUENCE
{                       confirmation               [1]   BOOLEAN DEFAULT FALSE,
                        ...
}

TP-END-DIALOGUE-RC  ::=  SEQUENCE
{                       ...
}

TP-U-ERROR-RI  ::=  SEQUENCE
{                       ...
}

TP-U-ERROR-RC  ::=  SEQUENCE
{                       ...
}
```

```
TP-ABORT-RI ::= SEQUENCE
{           type  CHOICE
                  {user                    [1]   SEQUENCE
                                                  {...,
                                                   ...,
                                                   user-data       [30]  User-information OPTIONAL
                                                  },
                   provider                [2]   SEQUENCE
                                                  {diagnostic      [1]   ENUMERATED
                                                       {permanent-failure         (1),
                                                        begin-transaction-reject  (2),
                                                        transient-failure         (3),
                                                        protocol-error            (4),
                                                        ...
                                                       },
                                                   ...
                                                  }
                  }
}

TP-GRANT-CONTROL-RI ::= SEQUENCE
{                      ...
}

TP-REQUEST-CONTROL-RI ::= SEQUENCE
{                      ...
}
TP-HANDSHAKE-RI ::= SEQUENCE
{                 confirmation-urgency     [1]   Confirmation-urgency OPTIONAL,
                       ...
}

TP-HANDSHAKE-RC ::= SEQUENCE
{                      ...
}

TP-HANDSHAKE-AND-GRANT-CONTROL-RI ::= SEQUENCE
{                 confirmation-urgency     [1]   Confirmation-urgency
                                                  DEFAULT  urgent,

                       ...
}

TP-HANDSHAKE-AND-GRANT-CONTROL-RC ::= SEQUENCE
{                      ...
}

TP-DEFER-RI ::= SEQUENCE
{                      type                [1]   ENUMERATED
                                                  {end-dialogue           (1),
                                                   grant-control          (2),
                                                   ...
                                                  } DEFAULT  end-dialogue,

                       ...
}

TP-PREPARE-RI ::= SEQUENCE
{                      data-permitted      [1]   BOOLEAN  OPTIONAL
                                                  -- present if polarized-control,

                       ...
}

TP-REPORT-RI ::= SEQUENCE
{                      heuristic-report    [1]   ENUMERATED
                                                  {heuristic-mix          (1),
                                                   heuristic-hazard       (2),
                                                   ...,
                                                   none                   (3)
                                                  }  DEFAULT   heuristic-mix,

                       ...,
                    severity                    [2]   ENUMERATED
                                                        {unknown              (0),
                                                        transient-specific    (1),
                                                         transient-general    (2),
                                                         permanent-specific   (3),
                                                         permanent-general    (4),
                                                         ...
                                                        } OPTIONAL,
                    diagnostic                  [3]   Diagnostic-code OPTIONAL,
                    extensions                  [4]   SEQUENCE
                                                        { ...
                        -- future extensions to TP-REPORT-RI that are to come before
                        -- the completion-data can be inserted here
                                                        } OPTIONAL,
                    completion-data             [30]  User-information OPTIONAL
}

TP-HEURISTIC-REPORT-RI ::= TP-REPORT-RI
            -- TP-HEURISTIC-REPORT-RI is a synonym for TP-REPORT-RI
            -- The name TP-HEURISTIC-REPORT-RI was used in the earlier editions of this
            -- specification and is retained for compatibility purposes only.

TP-TOKEN-GIVE-RI ::= SEQUENCE
{
                    reason                      [1]   ENUMERATED
                                                        {regular            (1),
                                                         keep               (2),
                                                         two-way-recovery   (3),
                                                         ...
                                                        } DEFAULT regular,
                    correlator                  [2]   Correlator    OPTIONAL,
                    ...
}

TP-TOKEN-PLEASE-RI ::= SEQUENCE
{                   ...
}

TP-RECOVER-RI ::= SEQUENCE
{                   recovery-context-handle     [1]   Recovery-context-handle,
                    ...
}

TP-INITIALIZE-RI ::= SEQUENCE
{                   protocol-version            [1]   Protocol-versions
                                                        DEFAULT {version1},
                    contention-winner-assignment [2]  BOOLEAN DEFAULT TRUE,
                        -- The value 'TRUE' means that the association initiator is
                        -- the contention-winner. The value 'FALSE' means that the
                        -- association acceptor is the contention-winner.
                    bid-mandatory               [3]   BOOLEAN DEFAULT TRUE,
                        -- The value TRUE means that the Bid mechanism must be used.
                        -- The value FALSE means that it may optionally be used.

                    recovery-context-handle     [4]   Recovery-context-handle
                                                        OPTIONAL,
                    ...,
                    functional-unit-capability  [5]   FU-list
                                DEFAULT {polarized-control,shared-control,
                                         commit-and-chained-transactions,
                                         commit-and-unchained-transactions,
                                         handshake,recovery)
                                        }
}
```

```
TP-INITIALIZE-RC  ::= SEQUENCE
{                       protocol-version            [1]   Protocol-versions DEFAULT {version1},
                        recovery-context-handle     [2]   Recovery-context-handle OPTIONAL,
                        diagnostic                  [3]   BIT STRING
                                    {ccr-version-2-not-available                (0),
                                     tp-protocol-version-incompatibility        (1),
                                     contention-winner-assignment-rejected      (2),
                                     bid-mandatory-value-rejected               (3),
                                     no-reason-given                            (4)
                                    } OPTIONAL,
                           -- the field is not present if the BIT STRING value is empty.
                        ...,
                        functional-unit-capability  [5]   FU-list
                                 DEFAULT {polarized-control,shared-control,
                                          commit-and-chained-transactions,
                                          commit-and-unchained-transactions,
                                          handshake,recovery)
                                         }
}

TP-BEGIN-TRANSACTION-RI  ::=  SEQUENCE
{                       check-ready-directions      [1]   Check-ready-directions
                                                             DEFAULT FALSE,

                        ...
}

TP-NEXT-TID-RI  ::=  SEQUENCE
{                       next-transaction-identifier [0]   TRANSACTION-IDENTIFIER,
                        next-branch-suffix          [1]   BRANCH-SUFFIX,
                        ...
}

TP-ABORT-AND-REPORT-RI  ::=  SEQUENCE
{                          heuristic-report         [1]   ENUMERATED
                                                             {heuristic-mix       (1),
                                                              heuristic-hazard    (2),
                                                              ...,
                                                              none                (3)
                                                             }  DEFAULT   heuristic-mix,
                           severity                 [2]   ENUMERATED
                                                             {unknown             (0),
                                                              transient-specific  (1),
                                                              transient-general   (2),
                                                              permanent-specific  (3),
                                                              permanent-general   (4),
                                                              ...
                                                             } OPTIONAL,
                           diagnostic               [3]   Diagnostic-code OPTIONAL,
                           ...,
                           ...,
                           user-data                [29]  User-information OPTIONAL,
                           completion-data          [30]  User-information OPTIONAL
}

TP-SOLICIT-DIALOGUE-RI  ::=  SEQUENCE
{                   last-partner-identifier         [1]   Correlator OPTIONAL,
                    candidate-initiating-tpsu-titles [2]  SEQUENCE OF TPSU-title
                                                             OPTIONAL,
                    candidate-responding-tpsu-titles [3]  SEQUENCE OF TPSU-title
                                                             OPTIONAL,

                    ...
}

TP-SOLICIT-DIALOGUE-RC  ::=  SEQUENCE
{                   ...
}
```

```
-- Supporting type definitions --

TRANSACTION-IDENTIFIER  ::= SEQUENCE
                                        {owners-name CHOICE
                                             {name                        [0]   EXPLICIT AE-title,
                                              side                        [1]   ENUMERATED
                                                                                     { superior      (0),
                                                                                       subordinate   (1),
                                                                                       ...
                                                                                     },
                                              ...
                                             },
                                        suffix              CHOICE
                                             {form1                       [2]   OCTET STRING,
                                              form2                       [3]   INTEGER,
                                              ...
                                             }
                                   }
               -- This type definition is syntactically identical to the
               -- ATOMIC-ACTION-IDENTIFIER in
               -- CCR-2 {joint-iso-itu-t ccr(7) module(1) ccr-apuds1(1) version3(3).
               -- It is a sender's option whether, if the owners-name is the
               -- AE-title of the superior or subordinate of the dialogue that
               -- originally supported this transaction branch, the "name" form
               -- or the appropriate "side" form is used.

BRANCH-SUFFIX  ::=   CHOICE
                                        {form1              OCTET STRING,
                                         form2              INTEGER,
                                         ...
                                        }
               -- This type definition is syntactically identical to the
               -- BRANCH-SUFFIX in
               -- CCR-2 {joint-iso-itu-t ccr(7) module(1) ccr-apuds1(1) version3(3).

BRANCH-IDENTIFIER  ::=         SEQUENCE
                                        {branch-owners-name               CHOICE
                                             {name                        [0]   EXPLICIT AE-title,
                                              side                        [1]   ENUMERATED
                                                                                     { superior      (0),
                                                                                       subordinate   (1),
                                                                                       ...
                                                                                     },
                                              ...
                                             },
                                         suffix             CHOICE
                                             {form1         [2]   OCTET STRING,
                                              form2         [3]   INTEGER,
                                              ...
                                             }
                                        }
               -- This type definition is syntactically identical to the
               -- BRANCH-IDENTIFIER in
               -- CCR-2 {joint-iso-ccitt ccr(7) module(1) ccr-apuds1(1) version3(3).
               -- It is a sender's option whether, the "name" form
               -- or the appropriate "side" form is used for the
               -- branch-owners-name.
               -- (The branch-owners-name will always be the superior or
               -- subordinate of the original dialogue.)

Check-ready-directions  ::=  BOOLEAN

Confirmation-urgency     ::=  ENUMERATED
                                   {urgent          (1),
                                    normal          (2)
                                   }
```

```
Diagnostic-code          ::=  INTEGER
                                {user-rollback (1),
                                 user-data-transaction-completion-collision (2),
                                 early-exit-completion-collision (3),
                                 other-provider-rollback (4),
                                 user-protocol-error (5)
                                }
                  -- Implementations shall accept other positive values for
                  -- the Diagnostic-code.

Correlator ::= INTEGER
                              -- unique within the scope of the association

FU-list                       ::=     BIT STRING
                                      {polarized-control                           (0),
                                       shared-control                              (1),
                                       commit-and-chained-transactions             (2),
                                       commit-and-unchained-transactions           (3),
                                       handshake                                   (4),
                                       recovery                                    (5),
                                       dynamic-commitment                          (6),
                                       unchecked-tree                              (7),
                                       implicit-prepare                            (8),
                                       read-only                                   (9),
                                       one-phase-commit-and-chained-transactions   (10),
                                       one-phase-commit-and-unchained-transactions (11),
                                       completion-diagnostics                      (13),
                                       heuristic-containment-required              (14),
                                       rch-on-dialogue                             (15),
                                       cancel                                      (16),
                                       solicit-dialogue                            (17)
                                      }

Protocol-versions             ::=  BIT STRING       {version1(0)}

Ready-flow-controls ::=       BIT STRING
                                          { Subordinate-may-send-ready     (0),
                                            Superior-may-send-ready        (1)
                                          }

Recovery-context-handle       ::=  OCTET STRING

TPSU-title                    ::=  CHOICE
                                      {T61String,         -- May be used as an AttributeValue
                                      PrintableString,    -- for an RDN in a Directory Name.
                                      INTEGER
                                      }

User-information              ::=  SEQUENCE OF EXTERNAL

END -- of TPASE definitions
```

12.2 Rules of extensibility

To provide for future compatibility, for the TP-INITIALIZE-RI/RC and TP-BEGIN-DIALOGUE-RI/RC APDUs, a receiving TPPM shall ignore a received field or field value that is not defined within the APDU in the ASN.1 description of this version of this Recommendation. Where named bits are used for a BIT STRING type in the ASN.1 description, a receiving implementation shall treat any bit as insignificant when no name is assigned to it.

An APDU that is not defined in the ASN.1 description of the negotiated version of this Recommendation shall not be sent. If such an APDU is received, it shall be treated as a protocol error.

A received field or field value that is not defined within an APDU in the ASN.1 description of the negotiated version of this Recommendation (other than fields or field values of TP-INITIALIZE-RI/RC and TP-BEGIN-DIALOGUE-RI/RC APDUs) shall either be ignored or treated as a protocol error.

The abstract syntax name may be used when new fields or field values are defined within any TPASE-APDU.

13 Conformance

13.1 Static conformance requirements

13.1.1 Conformance classes

13.1.1.1 General requirements

The system shall support one or more of the following conformance classes:

a) Application Transaction Branches Class;

b) Unchained One-phase Commit Transaction Branches Class;

c) Chained One-phase Commit Transaction Branches Class;

d) Unchained Provider-Supported Transaction Branches Class;

e) Chained Provider-Supported Transaction Branches Class.

Support for conformance classes is defined in terms of support for functional units in the subclauses of 13.1.1, "Conformance Classes". Support for each functional unit is defined in terms of support for sending and/or accepting particular APDUs as defined in 13.1.2, "Capabilities" and 13.1.3, "Functional Units".

13.1.1.2 Application transaction branches class

A system in conformance with the Application Transaction Branches Class shall support:

a) the requirements of the Dialogue functional unit;

b) the Shared Control functional unit or the Polarized Control functional unit, or both; and

c) optionally, the Handshake functional unit.

13.1.1.3 Chained One-phase Commit Transaction Branches Class

A system in conformance with the Chained One-phase Commit Transaction Branches Class shall support:

a) the requirements of the Dialogue functional unit;

b) the Shared Control functional unit, or the Polarized Control functional unit, or both;

c) the One-phase Commit functional unit and this shall be selectable independently of the Commit functional unit if this is also selectable;

d) the Chained Transactions functional unit;

e) optionally, the Implicit Prepare functional unit;

f) optionally, the Read Only functional unit;

g) optionally, the Completion Diagnostics functional unit;

h) optionally, the Heuristic Containment Required functional unit;

i) optionally, the Cancel functional unit; and

j) optionally, the Handshake functional unit.

13.1.1.4 Unchained One-phase Commit Transaction Branches Class

A system in conformance with the Unchained One-phase Commit Transaction Branches Class shall support:

a) the requirements of the Dialogue functional unit;

b) the Shared Control functional unit, or the Polarized Control functional unit, or both;

c) the One-phase Commit functional unit and this shall be selectable independently of the Commit functional unit if this is also selectable;

d) the Unchained Transactions functional unit;

e) optionally, the Implicit Prepare functional unit;

f) optionally, the Read Only functional unit;

g) optionally, the Completion Diagnostics functional unit;

h) optionally, the Heuristic Containment Required functional unit;

i) optionally, the Cancel functional unit; and

j) optionally, the Handshake functional unit.

13.1.1.5 Chained provider-supported transaction branches class

A system in conformance with the Chained Provider-Supported Transaction Branches Class shall support:

a) the requirements of the Dialogue functional unit;

b) the Shared Control functional unit, or the Polarized Control functional unit, or both;

c) the Commit functional unit;

d) the Chained Transactions functional unit;

e) optionally, the One-phase Commit functional unit;

f) optionally, the Implicit Prepare functional unit;

g) optionally, the Read Only functional unit;

h) optionally, the Completion Diagnostics functional unit;

i) optionally, the Heuristic Report Suppression functional unit;

j) optionally, the Cancel functional unit;

k) the Recovery functional unit;

l) optionally, the RCH on dialogue functional unit; and

m) optionally, the Handshake functional unit.

13.1.1.6 Unchained provider-supported transaction branches class

A system in conformance with the Unchained Provider-Supported Transaction Branches Class shall support:

a) the requirements of the Dialogue functional unit;

b) the Shared Control functional unit, or the Polarized Control functional unit, or both;

c) the Commit functional unit;

d) the Unchained Transactions functional unit;

e) optionally, the Dynamic Commit functional unit;

f) optionally, the One-phase Commit functional unit;

g) optionally, the Implicit Prepare functional unit;

h) optionally, the Read Only functional unit;

i) optionally, the Completion Diagnostics functional unit;

j) optionally, the Heuristic Containment Required functional unit;

k) optionally, the Cancel functional unit;

l) optionally, the Unchecked Tree functional unit;

m) the Recovery functional unit;

n) optionally, the RCH on dialogue functional unit; and

o) optionally, the Handshake functional unit.

13.1.2 Capabilities

13.1.2.1 General capabilities

The system shall be capable of:

a) Initiating the establishment of an application association (by sending a TP-INITIALIZE-RI APDU and receiving a TP-INITIALIZE-RC APDU) (role "Ai"), or

accepting the establishment of an association (by receiving a TP-INITIALIZE-RI APDU and sending a TP-INITIALIZE-RC APDU) (role "Aa"), or

both initiating and accepting the establishment of an association (roles Ai and Aa).

b) Functioning as a contention-winner of an association (role "Cw"), or

functioning as a contention-loser of an association (role "Cl"), or

functioning as both a contention-winner and contention-loser (roles "Cw" and "Cl").

c) Initiating a TP dialogue (role "Di"), or

accepting a TP dialogue (role "Da"), or

both initiating and accepting a TP dialogue (roles "Di" and "Da").

d) When the Commit functional unit is supported,

initiating a transaction branch (role "Ti"), or

accepting a transaction branch (role "Ta"), or

both initiating and accepting a transaction branch (roles "Ti" and "Ta").

e) When the Recovery functional unit is supported, the system shall have both capabilities, role Ai and Aa, stated in 13.1.2.1a) above for the purpose of recovery;

NOTE – The following are required capabilities of all systems regardless of role.

f) Rejecting a TP dialogue.

g) Supporting the rules of extensibility specified in the "Rules of Extensibility" subclause (12.2).

13.1.2.2 Constraints

A system in conformance with the role of initiating a transaction branch shall support initiating a TP dialogue with the Commit functional unit selected.

A system in conformance with the role of accepting a transaction branch shall support accepting a TP dialogue with the Commit functional unit selected.

13.1.3 Functional units

13.1.3.1 Definition

TP functional units are logical groupings of related TP protocol elements.

The TP functional units comprise:

a) the functional units visible to the TP service, as defined in ITU-T Rec. X.861 | ISO/IEC 10026-2;

b) the Rch-on-dialogue, Cancel, and Recovery functional units, as defined below.

13.1.3.2 Description of Rch-on-dialogue functional unit

The Rch-on-dialogue functional unit is internal to the TPPM and CPM and therefore is not directly accessible to any TPSUI.

The Rch-on-dialogue functional unit provides protocols that allow a TPPM to specify the Recovery Context Handle at dialogue establishment, overriding any value of Recovery Context Handle specified at association establishment.

13.1.3.3 Description of Cancel functional unit

The Cancel functional unit is internal to the TPPM and therefore is not directly accessible to any TPSUI.

The Cancel functional unit provides protocols that allow a rollback to be communicated to a superior as soon as it is known to the subordinate, without waiting for replies from lower subordinates.

13.1.3.4 Description of Recovery functional unit

The Recovery functional unit is only used on a TP channel. It is internal to the TPPM and the CPM and therefore is not directly accessible to any TPSUI.

The Recovery functional unit provides the protocols necessary to allow a CPM to send a request for, accept a request for, reject a request for, normally terminate and abnormally terminate a TP channel.

The Recovery functional unit also provides the protocol necessary to allow a TPPM to effect the rollback or commitment of transactions that have been affected by a failure.

13.1.3.5 Requirements on TP APDUs

Table 51 shows whether support for sending/receipt of an APDU is necessary to support a given functional unit. When an APDU shall be supported, in the context of the functional unit where it appears, independently of any capability, then it is marked "M" (mandatory).

Where the requirements depend on the capabilities that are supported (see 13.1.2.1), the two-letter mnemonics identifying the roles (e.g. Ai) are used: the notation "Xy" means "The sending (or receipt) of the APDU shall be supported if the capability identified by Xy is supported."

More complicated conditions are spelled out beneath the table. They are referred to in the table using the notation "(Cn)".

Where the system supports functional units other than the *Basic functional units*, the relationships between the functional units supported, capabilities and the requirements for support of particular APDUs and APDU fields are not fully expressed in Table 51. The requirements in these cases are defined in the PICS proforma provided in ITU-T Rec. X.863 | ISO/IEC 10026-4, Annex A, which uses the notation defined in CCITT Rec. X.292 | ISO/IEC 9646-3. The notation "(Cp)" is used in Table 51 to indicate that the support requirement for the APDU, if functional units other than *Basic functional units* are supported, is defined in the PICS proforma.

A system that does not fulfill a condition expressed in the support column is not required to be capable of sending or receiving the corresponding TP APDUs.

Where a TP APDU appears in both the Dialogue and the Recovery functional units, a system that supports both functional units shall meet the requirements of both sets of conditions.

NOTE – Implementations supporting only *Basic functional units* that are restricted in that they do not send all APDUs required by this subclause may be reasonable in specific application environments. Currently, these implementations are not conforming to this Recommendation. However, it is under study as to whether such implementations in the future may claim conformance to the TP Protocol. This Note does not modify any part of clause 13.

13.1.4 Dependencies on other standards

The system shall also implement the following standards:

a) The CCR protocol in conformance with ITU-T Rec. X.852 | ISO/IEC 9805-1, if the Commit functional unit is supported.

   The system shall support the role of a CCR branch-initiator if the system supports role Ti, and the system shall support the role of a CCR branch-responder if the system supports role Ta.

   If the system supports both roles Ti and Ta, then the system shall support the combination of the roles of CCR branch-initiator and CCR branch-responder as specified in the TP procedures (see clauses 9, 10, and 11).

   If the system supports the TP Dynamic Commit functional unit, it shall support the CCR Dynamic Commitment Functional unit, as specified in ITU-T Rec. X.852 | ISO/IEC 9805-1.

If the system supports either or both of the TP Read-only or TP One-phase Commitment functional units, it shall support the CCR No-change Functional unit, as specified in ITU-T Rec. X.852 | ISO/IEC 9805-1.

If the system supports the TP Cancel functional unit, it shall support the CCR Cancel Functional unit, as specified in ITU-T Rec. X.852 | ISO/IEC 9805-1;

b) The ACSE protocol in conformance with ITU-T Rec. X.227 | ISO/IEC 8650-1.

The system shall support the normal mode.

The system shall support the role of association initiator if the system supports role Ai and shall support the role of association responder if the system supports role Aa;

c) The Presentation protocol in conformance with ITU-T Rec. X.226 | ISO/IEC 8823-1.

In addition to those services used by ACSE, the system shall support P-DATA service primitives.

If the Commit functional unit is supported, then, in addition to those services used by CCR, the system shall support the use of the P-TOKEN-GIVE (synchronize-minor) service primitives. In addition, when the system supports the Recovery functional unit and uses the two-way-recovery facility, the system shall also support the P-TOKEN-PLEASE (synchronize-minor) service primitives.

d) The ASN.1 basic encoding rules in conformance with ITU-T Rec. X.690 | ISO/IEC 8825-1 (even if the system supports other encodings);

e) The Session protocol in conformance with ITU-T Rec. X.225 | ISO/IEC 8327-1.

The system shall support version 2.

The system shall support the Kernel and Duplex functional units.

13.2 Dynamic conformance requirements

13.2.1 General requirements

a) the system shall correctly generate, accept, and respond to all valid protocol elements that support each class to which conformance is claimed;

b) the system shall respond to all incorrect sequences of TP protocol elements.

13.2.2 Specific requirements

For each conformance class to which conformance is claimed and for each option of the static conformance requirements implemented, the system shall exhibit external behaviour consistent with having implemented the following:

a) a TP Protocol Machine as specified in the "TP-ASE description", "SACF description", and "MACF description" (clauses 9, 10, and 11) interpreted in accordance with the "Execution rules" (clause 7);

b) the association management functions defined in the "Association management" (8.5);

c) the use of the Association Control Service Element and of the Presentation Layer, as specified in 8.2, "Use of ACSE Service Primitives", and 8.4, "Use of the Presentation Layer";

d) encoding of TP APDUs as specified in 12.1, "Abstract Syntax of the TP-ASE APDUs";

e) embedding of APDUs as described in 6.1.10, "Embedding";

f) separation of APDUs as described in 6.1.9, "Concatenation/separation".

13.3 Protocol Implementation Conformance Statement

The supplier of a protocol implementation for which conformance to this Recommendation is claimed shall complete a copy of the PICS proforma provided in ITU-T Rec. X.863 | ISO/IEC 10026-4, Annex A, and shall provide the information necessary to identify both the supplier and the implementation.

NOTE – The concepts of a PICS and a PICS proforma are defined in ITU-T Rec. X.290 | ISO/IEC 9646-1.

Table 51/X.862 – TP Functional Units and support for TP APDUs

Functional Units	TP APDU	Support	
		Sending	Receiving
Dialogue	TP-BEGIN-DIALOGUE-RI ("dialogue" structure)	Di	M
	TP-BEGIN-DIALOGUE-RC (accept)	Da	Di
	TP-BEGIN-DIALOGUE-RC (reject)	M	Di
	TP-END-DIALOGUE-RI	(C1)	(C1)
	TP-END-DIALOGUE-RC	(C1)	(C1)
	TP-U-ERROR-RI	M	M
	TP-ABORT-RI	M	M
	TP-BID-RI	(C5)	Cw
	TP-BID-RC	Cw	(C2)
	TP-INITIALIZE-RI	Ai	Aa
	TP-INITIALIZE-RC	Aa	Ai
Shared Control	TP-U-ERROR-RC	M	M
Polarized Control	TP-GRANT-CONTROL-RI	M	M
	TP-REQUEST-CONTROL-RI	M	M
Handshake	TP-HANDSHAKE-RI	M	M
	TP-HANDSHAKE-RC	M	M
Handshake and Polarized Control	TP-HANDSHAKE-AND-GRANT-CONTROL-RI	M	M
	TP-HANDSHAKE-AND-GRANT-CONTROL-RC	M	M
Commit	TP-PREPARE-RI	Ti(Cp)	Ta(Cp)
	TP-DEFER-RI	Ti	Ta
	TP- REPORT-RI	Ta(Cp)	Ti(Cp)
	TP-TOKEN-GIVE-RI	M	M
Unchained Transactions	(None)		
Chained Transactions	(None)		
Recovery	TP-BEGIN-DIALOGUE-RI ("channel" structure)	M	M
	TP-BEGIN-DIALOGUE-RC	M	M
	TP-BID-RI	(C5)	Cw
	TP-BID-RC	Cw	(C2)
	TP-RECOVER-RI	M	(C4)
	TP-TOKEN-PLEASE-RI	(C3)	(C3)
	TP-END-DIALOGUE-RI	M	M
	TP-INITIALIZE-RI	M	M
	TP-INITIALIZE-RC	M	M
Dynamic Commit	TP-PREPARE-RI	(Cp)	(Cp)
Unchecked Tree	TP-BEGIN-TRANSACTION-RI	Ti	Ta
Implicit Prepare	(None)		
Read Only	(None)		
Early-exit	TP-EARLY-EXIT-RI	Ti	Ta
	TP-EARLY-EXIT-RC	Ta	Ti
One-phase Commit	TP-ONE-PHASE-RI	(Cp)	(Cp)
Completion Diagnostics	TP-REPORT-RI	Ti	Ta
Heuristic Containment Required	(None)		
Cancel	(None)		
Rch-on-dialogue	(None)		

(C1) The sending and receipt of both the TP-END-DIALOGUE-RI and TP-END-DIALOGUE-RC shall be supported when one, or both, of the Application Transaction Branches class or the Unchained Provider-supported Transaction Branches class are supported.

(C2) The receipt of the TP-BID-RC APDU shall be supported when the system is capable of sending the TP-BID-RI APDU.

(C3) Both the sending and the receipt of the TP-TOKEN-PLEASE-RI APDU shall be supported when the two-way-recovery facility is used.

(C4) The receipt of the TP-RECOVER-RI APDU shall be supported when the system supplies a recovery-context-handle on associations which it will use for initiating or accepting provider-supported transactions.

(C5) Sending TP-BID-RI is optional. To use a given association for either a dialogue or a channel, it may be necessary to bid. See clauses 8 and 10 for the specific circumstances under which bidding is required to use a particular association.

13.4 Receiving TP APDUs

The semantics of some optional fields of some TP APDUs only apply to certain conformance options.

The system shall accept the syntax of all validly formatted fields in received TP APDUs. However, the system may ignore the semantics of those fields referred to in the first sentence of this subclause, if the system does not support the corresponding static conformance options.

14 Compliance

This Recommendation complies with the service user rules of CCR, ITU-T Rec. X.851 | ISO/IEC 9804.

15 Precedence statement

The text of clauses 7 through 12 takes precedence over the description contained in Annex A.

Predicates, variables, and states in the state tables reflect both the text procedures (defined in the previous clauses) and the sequencing rules defined in ITU-T Rec. X.861 | ISO/IEC 10026-2. The text procedures are augmented with certain Service rules (see 7.2) to provide the same detection of illegal behaviour.

16 Index of Actions and Events

Annex A

OSI TP Protocol – State tables

A.1 General

This annex describes the OSI TP Protocol in terms of state tables. The state tables show the state of both the TPPM and the CPM, the events that occur in the OSI TP Protocol, the actions taken, and the resulting states.

A.2 Introduction

A.2.1 State tables

The TP state tables used in the description of the OSI TP Protocol comprise:

a) the TPPM MACF state table, presented as six tables:

- A.13 Dialogue;
- A.14 Handshake;
- A.15-1 Commitment – Subordinate;
- A.15-2 Commitment – Superior;
- A.15-3 Zombie and Node related finite state machine;
- A.16 Rollback.

NOTE 1 – The TPPM MACF state table is presented as six tables because of the difficulty of presenting all the states on a page.

b) the CPM MACF state table (Table A.17);

c) the SACF state table (Table A.18); and

d) the TPASE table (Table A.11).

NOTE 2 – The TPASE (Table A.11) encodes and decodes TP APDUs.

A.2.2 PM state machines

A.2.2.1 PM instance

A state machine is an instance of a state table. An instance of a PM consists of several instances of the state machines described in the following subclauses. The state machines that comprise a PM instance cooperate by the exchange of events (see A.2.3), the use of shared variables, and the counting mechanism [see A.3 rule f)].

A.2.2.2 TPPM

The state of a TPPM at any particular time is represented by the state in each of the:

a) TPPM MACF state machines. There is one state machine per dialogue branch or transaction branch, depending on the coordination level; and

b) SAO state machines. There is one SAO state machine per SAO in use by the TPPM, comprising:

1) an SACF state machine;
2) a TPASE state machine;
3) a CCRPM (refer to ITU-T Rec. X.852 | ISO/IEC 9805-1), if CCR is in the application context;
4) an ACPM (refer to ITU-T Rec. X.227 | ISO/IEC 8650-1); and
5) one or more U-ASE state machines.

The TPPM state consists of the combined set of states of each of the above state machines as well as additional context related to the entire TPPM.

NOTE – An example of this context is the node variables defined in A.2.5.

The TPPM MACF states are described in A.4.1.1. The SACF states are described in A.6.1.

A.2.2.3 CPM

The CPM MACF as described in 6.2.1 is represented by a CPM MACF state machine for each channel.

The state of a channel within the CPM at any particular time is represented by the state in each of the:

a) CPM MACF state machine; and

b) SAO state machine. There is one SAO state machine per SAO in use by the CPM, comprising:

   1) an SACF state machine;

   2) a TPASE state machine;

   3) a CCRPM (refer to ITU-T Rec. X.852 | ISO/IEC 9805-1); and

   4) an ACPM (refer to ITU-T Rec. X.227 | ISO/IEC 8650-1)

The CPM MACF states are described in A.4.1.2. The SACF states are described in A.6.1.

A.2.2.4 The Node related Finite State Machine (NFSM)

Each MACF contains exactly one instance of a node related finite state machine table, the Node related Finite State Machine (NFSM). The NFSM is used to complete the service boundary sequence of events, if there is no coordinated dialogue and therefore no dialogue related finite state machine with Dl = TRUE.

The NFSM is either in the awake state (if a transaction identifier is in use at the node, i.e. a transaction branch for this transaction exists or has existed and this transaction is not yet completed) or in the dormant state otherwise.

The NFSM is woken up, i.e. enters the awake state, when:

1) a transaction identifier is received on the branch with the superior; or

2) a new transaction identifier is created at the node; or

3) recovery is initiated at the node after a node crash.

The NFSM is deactivated, i.e. enters the dormant state, when:

1) the MACF is newly created (i. e. the NFSM is created in the dormant state);

2) the dialogue/branch with the superior is rejected;

3) the current transaction is completed without initiation of a new transaction.

The NFSM in the awake state follows the main state transitions of a transaction at a node.

The node stays in the awake state if a next transaction is initiated during the completion of the current transaction, i.e. a new transaction identifier is in use at the node.

The state of the NFSM is initialized as follows:

1) the initial main state is 26.1;

2) the variable Dl is set to TRUE (to enable synchronizing events);

3) all other variables are set to their initial values as given in A.4.2.4.

A.2.3 Events

PM **input events** are defined in 7.1.2.

State machines are affected by **incoming events** which comprise PM input events, internal events and synchronizing events. As a result of processing a single incoming event, a state machine may generate zero or more **outgoing events**, some of which may become incoming events to another state machine.

NOTE 1 – The following examples illustrate how events are received and generated by the state machines. Refer to Figure A.1.

Example 1 – The state of the TPPM MACF state machine is assumed to be in state 1. Input event TP-BEGIN-DIALOGUE request is received by the TPPM MACF state machine (P1). The event is received according to Table A.13, state 1, and is processed by taking actions, among which outgoing event AF-BEGIN-DIALOGUE request is generated to the SACF state machine (P4).

Incoming event AF-BEGIN-DIALOGUE request is received by the SACF state machine (P4) according to Table A.18, state 1. As a result, the SACF state machine issues an outgoing event which is either an AF-BEGIN DIALOGUE request or an AF-BID request to the TPASE (P5). The TPASE encodes the AF-Service request into a TP APDU, according to Table A.11, and generates an outgoing event: it issues (or the SACF may possibly concatenate with other TP APDUs – see 6.1.6) the corresponding Presentation-service request with the TP APDU as user-data (P9).

Example 2 – Input event P-DATA indication with a TP-BID-RI APDU as user-data is received from the PSAP (P3). The APDU is decoded by the TPASE according to Table A.11 and outgoing event AF-BID indication is generated to the SACF state machine (P10).

Incoming event AF-BID indication is received by the SACF state machine according to Table A.18. Assuming that it is accepted, the SACF machine generates outgoing event AF-BID response to the TPASE (P5). The TPASE encodes the AF-Service request into a TP APDU, according to Table A.11, and generates an outgoing event: it issues (or the SACF may possibly concatenate with other TP APDUs – see 6.1.6) the corresponding Presentation-service request with the TP APDU as User data (P9).

This second example shows an input event being completely handled by the TPASE and the SACF state machine without the involvement of the TPPM MACF state machine.

In addition, the present annex also uses the following event types.

An **internal event** is an event that is created as a result of some internal decision or occurrence (including internal or protocol errors). Internal events are described in A.4.3.1 and A.6.3.1.

A **synchronizing event** is an event used to convey node-related information across all MACF state machines with coordination level "commitment" or "one-phase commitment". A synchronizing event is generated as a result of processing a node-related state machine incoming event on a single branch (a single MACF state machine), and is a state machine incoming event to *all* MACF state machines (with coordination level "commitment" or "one-phase commitment"), including the state machine that generated the synchronizing event.

NOTE 2 – For example, upon the arrival of a *commit indication* on the superior dialogue, a *commit request* must be issued to each subordinate in the transaction tree from which a C-READY indication has been received. A C-NOCHANGE indication may be issued to each neighbour, from which a *ready-substitute indication* has been received. This is done by generating a synchronizing event to *all* TPPM MACF state tables with coordination level of "commitment" or "one-phase commitment"; where applicable, the TPPM MACF state machines (the ones representing subordinate dialogues or channels in this case) will take actions that include issuing the *commit request or substitute*.

A synchronizing event is generated only after the subcell (see A.2.8, Conventions) is completely processed, i.e. all actions are complete and the transition is made to the next state. If multiple synchronizing events are generated, they are generated (and processed) in sequence in the order requested and in the same action sequence [see A.3 rule a)].

A further distinction of events is whether the event is global or not. A **global event** is an event that is applied to all MACF state machines with the coordination level "commitment" or "one-phase commitment". The global events are TP-COMMIT request, TP-ROLLBACK request, TP-DONE request, all synchronizing events, and some internal events (see A.4.3.1 for the global internal events).

The order of processing by the affected state tables of a global event is arbitrary, but the rule of atomicity holds (see A.3).

Internal and synchronizing events are described in A.4.3 and A.6.3. All other state machine input events are TP-, ACSE-, CCR-, AF-, SAF-, CAF-, or U-ASE-Services.

A.2.4 States

A state machine is in one state at any given time. Upon initial creation, the state machines are all in state 1, except upon creation after node crash, in which case they are created in the appropriate state (as defined by A.4.4.5, "Actions after node crash").

MACF states are described in A.4.1 and SACF states in A.6.1.

A.2.5 Variables and predicates

Each state machine uses variables for keeping track of certain information, and uses variables and predicates as conditions and predicate expressions. The variables are of type Boolean, Integer, Octet String, and Record Types (whose names begin with "T", and are described in A.4.2.2).

There are six categories of variables:

a) dialogue variables, (whose names begin with "D"), which are specific to each TPPM MACF state machines;

b) channel variables (whose names begin with "C"), which are specific to each CPM MACF state machines;

c) node variables (whose names begin with "N"), which are specific to the TPPM MACF state machines. These variables are shared between all TPPM MACF state machines for a node;

d) system variables (whose names begin with "S"), which are specific to the TPPM and CPMMAF state machines. These variables are accessible to any TPPM or CPM MACF state machine for the system. These variables retain some of their elements in the event of a node crash;

   NOTE – Further details are given in A.4.2.3.

e) local decision variables (whose names begin with "Ld"), which represent local decisions and options. Successive evaluations of local decision variables may yield different values. This non-deterministic behaviour models potential changes in system resources and local strategies;

f) association variables (whose names begin with "A"), which are related to a particular association. These variables are used by the SACF, however, some of these variables are also shared with the TPPM or CPM MACF while the MACF is attached.

Predicates (whose names begin with "P"), are inspected by any state machine and represent conditions outside of the TPPM.

Boolean functions which operate over sets of records are used to update and test membership in the system variables. These are described in A.4.4.1.

The Dialogue, Channel, Node, System, and some Local decision variables are described in A.4.2. The Association variables, and some Local decision variables are described in A.6.2. The Predicates are described in A.7.

A.2.6 Actions

Actions are presented in the state tables' cells between brackets ("[ ]"). These actions are described in A.4.4, A.5.4, and A.6.4.

For each valid incoming event (see A.2.8, "Conventions" below), all applicable actions are taken. Actions with free-form names often have conditions embedded in them.

A.2.7 Notation

Incoming events are represented by their name, with one or more attributes, when required. Some specific attributes are represented by a predicate, as follows:

– AAI is a predicate which is the value of the *atomic-action identifier* parameter of the received service primitive; and

– BI is a predicate which is the value of the *atomic-action-branch identifier* parameter of the received service primitive.

States are represented by a number. The integer part of the state number refers to the corresponding TP-Service state.

Predicate expressions are noted in the form of a list of variable values and/or predicates separated by commas.

"^" means "not", and is applied to variables of type Boolean and to Predicates.

"=" means "equal to", and is applied to variables of type Integer and Octet String.

"^=" means "not equal to", and is applied to variables of type Integer and Octet String.

">" means "greater than", and is applied to variables of type Integer.

A.2.8 Conventions

In the state tables, the intersection of an incoming event (row) and a state (column) forms a cell.

A subcell is a subset of a cell enclosed in a box.

The elements of a subcell are the following (given in their order of appearance in the subcell):

a) optionally, a predicate expression;

b) zero or more actions; and

c) a resultant state.

When a predicate expression holds for all the subcells of a same column, it is indicated at the top of the column, and not repeated in the subcells of that column.

A blank cell, a non-existent cell for an event, or a cell with no subcells for which the evaluation of the predicate expressions is true, represents an invalid event (see A.2.9.3) for that state.

A cell with a subcell for which the evaluation of the predicate expressions is true, represents a valid event (see A.2.9.2) for that state.

Predicate expressions in a cell are such that only one or zero subcells in a cell applies.

When a service primitive contains parenthetical arguments, these are as described in 9.2, 10.3, and 11.2, augmented by the following additional arguments:

a) a service parameter (left argument) and its value (right argument), separated by an equals sign (=);

b) the words "transaction branch" and "no transaction branch", which indicate that the TP-BEGIN-DIALOGUE request has been specified with the Chained Transactions functional unit selected or with the Begin-Transaction parameter set to "true", or with the Begin-Transaction parameter either absent or set to "false", respectively;

c) the words "one-way-recovery" or "two-way-recovery", which indicate the value of the Channel-Utilization parameter;

d) the name of a functional unit selected on the dialogue or the channel, followed by the words "fu selected"; or

e) the words "sync-minor", which indicate the Session token value.

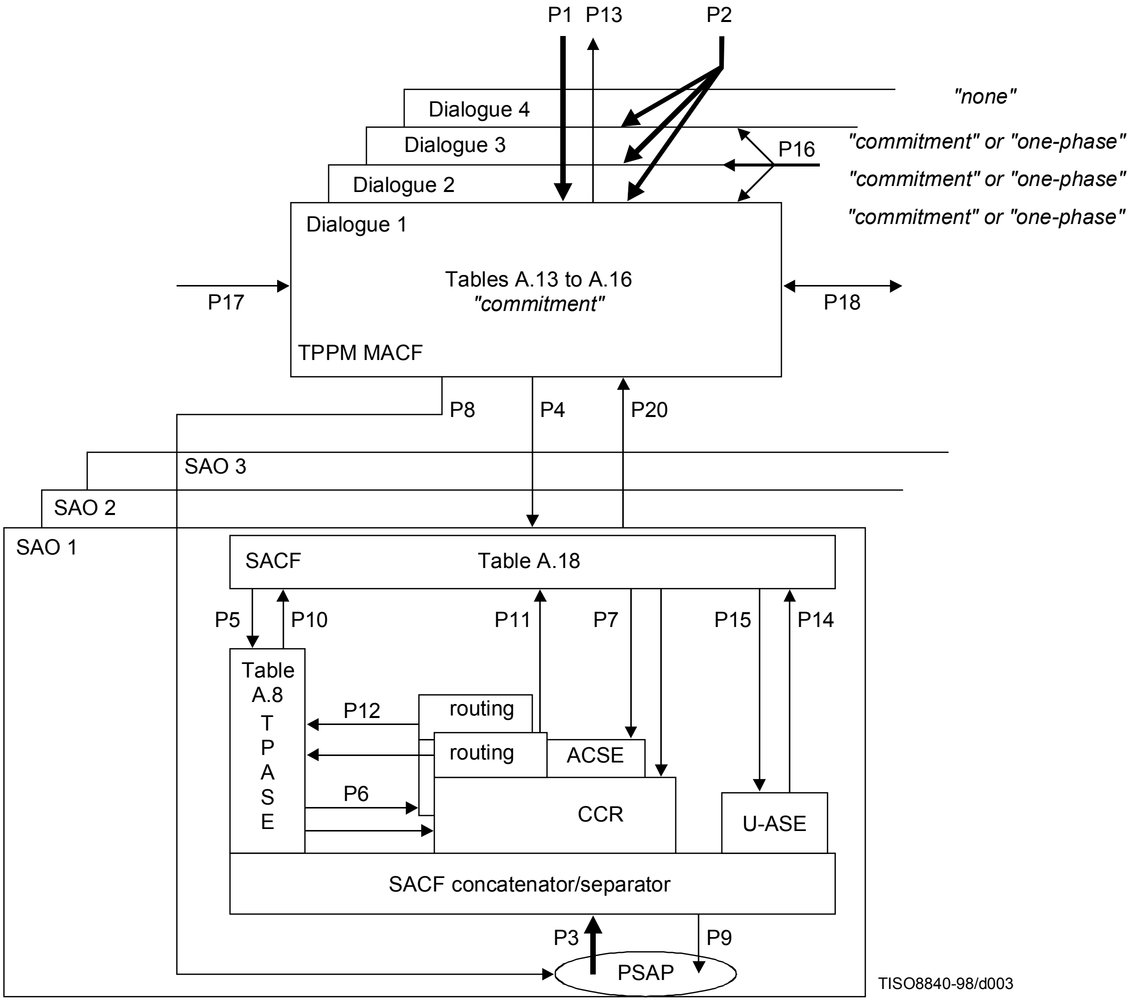
These arguments may appear in any location inside the parentheses.

A.2.9 Processing of events

A.2.9.1 Evaluating predicate expressions

An event is processed by evaluating predicate expressions in all the subcells of the cell for the current state. If any subcell's predicate expression evaluates true (e.g. "^Aw, Ldres" is the predicate expression and Aw is FALSE and Ldres is TRUE), or no predicate expression exists for the subcell, the event is valid for the event/state combination, and the actions are taken and the transition made.

The evaluation of predicate expressions does not have any side effect; in particular, local decision variables retain their value during the evaluation of predicate expressions appearing in subcells of a same cell.



P1 All TP- requests and responses except P2

P2 Global events: TP-COMMIT req, TP-ROLLBACK req, TP-DONE req, TP-ONE-PHASE req, TP-READ-ONLY req, TP-EARLY-EXIT req

P3 All incoming APDUs

P4 All AF- requests and responses (except AF-BID request and response), C- requests and responses, U-ASE request and all SAF-DETACH requests

P5 All AF- requests and responses

P6 All C- and A- requests and responses carrying a TP APDU as user data

P7 All C- and A- requests and responses without a TP APDU as user data

P8 Open/close PSAP

P9 All outgoing APDUs

P10 All AF- indications and confirms

P11 All C- and A- indications and confirms without a TP APDU as user data

P12 All A- and C- indications and confirms which contain TP APDUs

P13 All TP- indications and confirms

P14 U-ASE indication

P15 U-ASE request

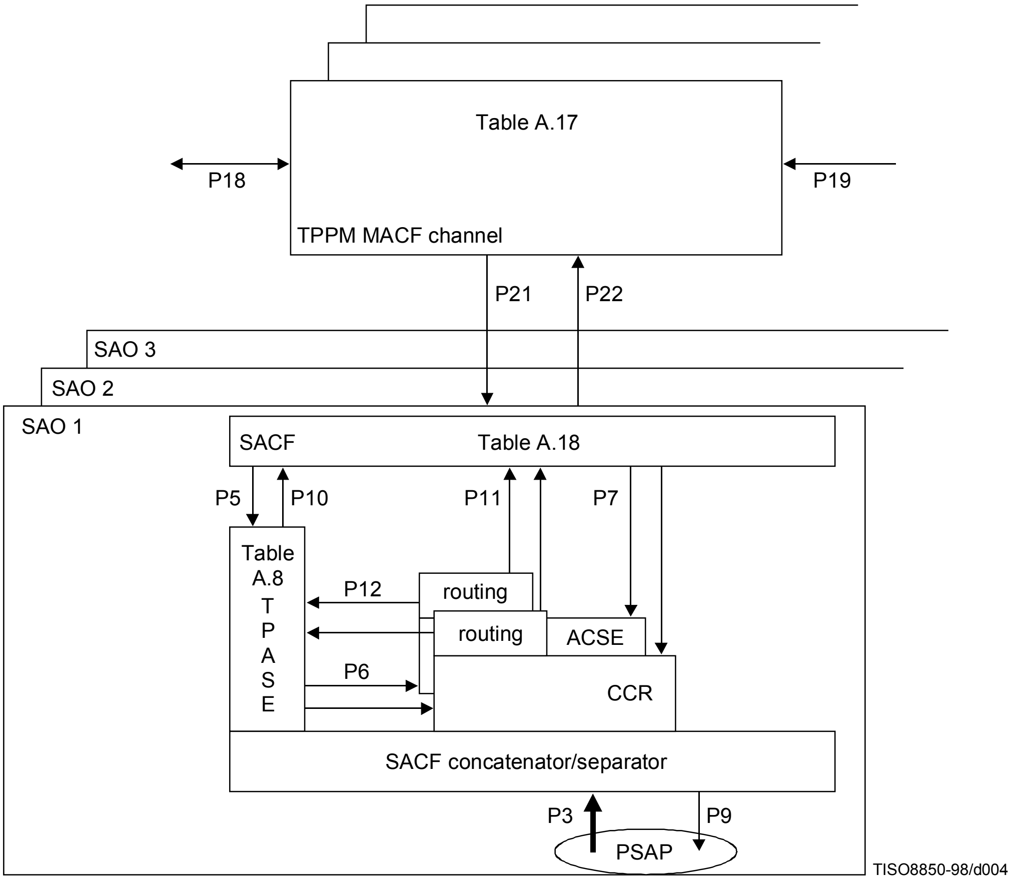
P16 All synchronizing events and global internal events

P17 TPPM Internal events

P18 CAF-services

P20 All AF-, C-, A-, and U-ASE indications and confirms, and SAF-ASSOCIATION-LOST indication

Figure A.1/X.862 – Flow of events through the TPPM state machines
(the NFSM is not shown)



P3	All incoming APDUs
P5	All AF- requests and responses
P6	All C- and A- requests and responses carrying a TP APDU as user data
P7	All C- and A- requests and responses without a TP APDU as user data
P9	All outgoing APDUs
P10	All AF- indications and confirms
P11	All C- and A- indications and confirms without a TP APDU as user data
P12	All A- and C- indications and confirms which contain TP APDUs
P18	CAF- services
P19	CPM Internal events
P21	AF-BEGIN-DIALOGUE and C-RECOVER requests and responses; AF-END-DIALOGUE, AF-RECOVER, AF-TOKEN-PLEASE, and AF-TOKEN-GIVE requests
P22	AF-BEGIN-DIALOGUE, and C-RECOVER indications and confirms; AF-END-DIALOGUE, AF-RECOVER, AF-TOKEN-PLEASE, and AF-TOKEN-GIVE indications

Figure A.2/X.862 – Flow of events through the CPM state machine

A.2.9.2 Processing valid events

For valid events, if the predicate expression (if any) is true, the following actions are taken:

a) the state machine performs the actions (if any) as shown in the cell; and

b) the state is changed to the specified resultant state (See also A.3, "Processing Rules", for additional rules in processing valid events).

A.2.9.3 Processing invalid events

Depending on the nature of the input events or state machine incoming events, one of the following actions is taken:

a) if the input event corresponds to the receipt of an invalid OSI TP Service primitive from the TPSUI, then, depending on a local decision, either the internal event "Internal error" is triggered or a node crash is triggered: see A.4.4.5 for actions after node crash (which corresponds to the procedures described in 7.1.6); or

b) if the state machine incoming event corresponds to receipt of an invalid APDU from the partner TPPM, the internal event "Protocol error" is triggered (which corresponds to the procedures described in 7.1.6).

A.3 Processing rules

The following rules complement the rules of normal processing of events described in A.2.9:

a) *Atomicity*

An input event is processed completely before any other input event is accepted. This means that any outgoing events created by actions that are state machine incoming events to other state machines are processed by those state machines, and so on, until the only unprocessed events are outgoing events which are not state machine incoming events (that is, they are events at the PSAP or TPSUI).

When processing a given input event, state machines may either execute in parallel provided exclusive access to variables which are shared between state machines is maintained, or they shall be executed serially.

b) *Routing*

When a service primitive received from the separator contains a TP APDU as user-data (or User Information in the case of ACSE), the service primitive becomes an event to the TPASE state machine.

When a service primitive received from the separator does not contain a TP APDU as user-data (or User Information in the case of ACSE), the service primitive becomes an event to the SACF state machine.

c) *Service conditions assumption*

The TPSUI is assumed to have issued requests and responses in accordance with the TPSUI conditions specified in ITU-T Rec. X.861 | ISO/IEC 10026-2.

NOTE 1 – The state tables enforce the TPSUI constraints as specified in ITU-T Rec. X.861 | ISO/IEC 10026-2.

d) *Context mechanism*

When a subcell is executed, the actions taken by the subcell are related either to the dialogue, if attached, or the channel if no dialogue is available and a channel is attached. In some cases, there is both a dialogue and a channel or two channels attached during a single action sequence. A context mechanism is provided to identify whether the actions should occur on the dialogue or on the previously existing channel. This mechanism is implemented by the actions DIALOGUE and OLDCHANNEL. If actions include the detaching of a channel, the subsequent actions will be taken on the remaining channel. This context switching mechanism works only within a single subcell.

e) *Channel assignment*

When a CAF-PLEASE request is issued by a TPPM state machine, a CPM state machine which can accept the event is either created in state 1 or found in another state if the AE-title of the channel is the same as the AE-Title parameter of the CAF-PLEASE request.

f) *Counting mechanism*

For all conditions that depend on a certain number of events occurring, a counting mechanism is used. The specific counters are set to the number of events which must occur to cause the node transition. Each time an event which is to be counted occurs, the counter is decremented. When the counter becomes zero, the one-time actions associated with the node are done by the state machine in which the counter becomes zero (see COUNTRDY, COUNTGE, COUNTCOM, and COUNTRB). These one-time actions include generation of the appropriate synchronizing events.

NOTE 2 – For example, when an intermediate node is to complete phase I of commitment, a *ready signal or ready-substitute indication* must have been received on each branch but one and *a transaction completion request* must have been received on each branch of the node. When the last of these events occurs, the synchronizing event "send-ready?" is generated, requesting the last branch to issue the *ready signal* or *ready-substitute request*. One counter (Ncntrdy) would be set to the number of branches (for the *ready signal* or *ready-substitute indication*). Another counter (Ncntge) would be set to the total number of branches (for the *transaction completion request*).

The counting mechanism is used in the following four cases:

- Counting the events necessary to complete the first phase of commitment. These events are:

  1) a *ready signal* or *ready-substitute indication* for each branch; and

  2) a *transaction completion request* for each branch.

  The counters Ncntrdy (for the *ready signal* or *ready-substitute indication*) and Ncntge (*transaction completion request*) are used. It is necessary to use two counters to distinguish between the two meanings of Ncnt = 1 (i.e. Ncntge + Ncntrdy = 1). If Ncntge = 1 then there is exactly one state machine having not processed the *transaction completion request*, but a *ready indication* or a *ready-substitute indication* has been received on each branch. The node may become the commit coordinator. If Ncntrdy = 1 then there is exactly one *ready indication* or *ready-substitute indication* missing and the node may send a *ready signal* or *ready-substitute request*.

- Counting the events necessary to complete the second phase of commitment. These events are:

  1) a *commit confirm* for each branch on which a *ready signal* has been received; and

  2) a TP-DONE request for each branch; and

  3) a C-BEGIN ind if there is a chaining superior dialogue and a *ready-substitute indication* has been received on the superior dialogue;

  4) an AF-REPORT(dataRI) indication or an AF-ABORT-AND-REPORT(dataRI) indication if the dialogue towards the coordinator is a subordinate dialogue and reporting applies on the dialogue.

  The counter Ncnt is used.

- Counting the events necessary to complete the rollback. These events are:

  1) a *rollback confirm* for each subordinate branch (except the subordinate branch from which a *rollback indication* has been received or is not awaited, if any); and

  2) a TP-DONE request for each branch.

  The counter Ncnt is used.

- Counting the events necessary for reporting on data to the superior.

  The counters Ncnthr and Ncntcr are used.

- Counting some specific dialogues at transaction branch establishment time:

  1) the number of two-phase-expected branches at the node;

  The counter Ncnt2exp is used.

  2) coordinated dialogues with one-phase functional unit selected;

  TP-ONE-PHASE request is not allowed if there is no coordinated dialogue with the one-phase functional unit selected.

  The counter Ncntopfu is used in procedure COUNTGE.

  3) coordinated dialogues with completion diagnostics functional unit selected;

  Completion Report parameters are only allowed with TP-ROLLBACK indication if there is at least one coordinated dialogue with completion report functional unit selected.

  The counter Ncntcdfu is used.

g) *Parameter inheritance*

As in the main text, the definition of parameter inheritance [see 7.2 a)] applies to the actions in this annex.

A.4 MACF state tables

A.4.1 MACF states

A.4.1.1 TPPM states

States are numbered with the following conventions:

a) the integer part of the state numbers corresponds to the state defined in ITU-T Rec. X.861 | ISO/IEC 10026-2;

b) states 2 through 8 and states 12 through 14 correspond to a node handling either an application supported transaction or a provider supported transaction in the ACTIVE state;

c) states 9 through 11 and 26.1 are specific to a node handling an application supported transaction;

d) states 15 through 20.2 and 26.2 are specific to a node handling a provider supported transaction in the ACTIVE state;

e) state 20.3 and 26.2 corresponds to a transaction node in the READY state;

f) state 20.3.2 and 26.2 corresponds to a transaction node in the ONE-PHASE state;

g) state 20.3.3 and 26.2 corresponds to a transaction node in the READ-ONLY state or the EARLY-EXIT state;

h) states 21.x and 26.3 correspond to a transaction node in the DECIDED (commit), DECIDED (commit-one-phase) or DECIDED (unknown) state; and

i) states 23.x and 26.4 correspond to a transaction node in the DECIDED (rollback) state.

The following states ared defined for the TPPM:

State 1

Idle state. No dialogue exists.

State 1.1

An AF-BEGIN-DIALOGUE indication has been received and the dialogue will have a coordination level of commitment. A C-BEGIN indication is awaited.

State 2

The TPSUI has control of the dialogue.

State 3

This state is valid only when the Polarized Control functional unit is selected. The dialogue is established and the TPSUI does not have control of the dialogue.

State 4

An AF-U-ERROR request was issued with the Shared Control functional unit selected or an AF-U-ERROR request was issued while the TPSUI did not have control of the dialogue.

State 5

This state is valid only when the Polarized Control functional unit is selected. The dialogue is established, the TPSUI has control of the dialogue, and an AF-U-ERROR indication has been received.

State 6

An AF-HANDSHAKE request has been issued. An AF-HANDSHAKE confirm is awaited.

State 7

An AF-HANDSHAKE indication has been received. A TP-HANDSHAKE response is awaited.

State 8

This state is valid only when both the Handshake and the Shared Control functional units are selected. An AF-HANDSHAKE indication has been received after an AF-HANDSHAKE request has been issued, or an AF-HANDSHAKE request has been issued, after an AF-HANDSHAKE indication has been received.

State 9

This state is valid only when both the Handshake and the Shared Control functional units are selected. An AF-END-DIALOGUE (*confirmation* = TRUE) indication has been received after an AF-HANDSHAKE request has been issued.

State 10

This state is valid only when both the Handshake and the Shared Control functional units are selected. An AF-HANDSHAKE indication has been received after an AF-END-DIALOGUE (*confirmation* = TRUE) request has been issued.

State 11

An AF-END-DIALOGUE (*confirmation* = TRUE) request has been issued. An AF-END-DIALOGUE confirm is awaited.

State 12

An AF-END-DIALOGUE (*confirmation* = TRUE) indication has been received. A TP-END-DIALOGUE response is awaited.

State 13

An AF-HANDSHAKE-AND-GRANT-CONTROL request has been issued. An AF-HANDSHAKE-AND-GRANT-CONTROL confirm is awaited.

State 14

An AF-HANDSHAKE-AND-GRANT-CONTROL indication has been received. A TP-HANDSHAKE-AND-GRANT-CONTROL response is awaited.

State 15

A TP-PREPARE request has been issued. A *ready signal or ready-substitute indication* is awaited. Unless the dynamic commit functional unit is selected, the dialogue must be *a subordinate dialogue*.

State 16.1

The TPSUI has issued a TP-PREPARE request and received a TP-PREPARE indication.

This state is used only for a dialogue with a coordination level of "commitment" and with the Dynamic Commit functional unit selected.

State 17

A *ready signal or ready-substitute indication* has been received. A *transaction completion request* is awaited. Unless the dynamic commit functional unit is selected, the dialogue must be *a subordinate dialogue*.

State 18

An AF-PREPARE indication has been received. A *transaction completion request* is awaited. Unless the dynamic commit functional unit is selected, the dialogue must be *a superior dialogue*.

State 20.1

A *transaction completion request* has been received. A *ready signal or ready-substitute indication* is awaited.

State 20.2

A *transaction completion request* and a *ready signal or ready-substitute indication* have been received. Synchronizing events "Continue-commit" (if a coordinator node) or "Enter-ready-state" are awaited.

State 20.3

The node is in the READY state. Synchronizing event "Enter-ready-state" has been received. A *commit indication* is awaited on the dialogue with the commit coordinator. Synchronizing event "Continue-commit" is awaited on all other dialogues. Recovery may be in progress on dialogues with subordinates.

If a *ready signal or ready-substitute indication* arrives on the dialogue with the commit coordinator, a collision of *ready signals or ready-substitute indication*s has occurred. The node may be the commit coordinator.

If recovery is in progress on the dialogue towards the commit coordinator and if a channel from the commit coordinator is attached – requesting the outcome of the transaction – the two *ready signals* are lost in a provider abort and a rollback is triggered. Both nodes remove the ready-logs.

State 20.3.2

The node is in the ONE-PHASE state. Synchronizing event "enter-one-phase-state" has been received.

A C-NOCHANGE confirmation is awaited on the dialogue towards the coordinator. Synchronizing event "Continue-commit" or "Continue-unknown" is awaited on all other dialogues.

If a *ready signal or ready-substitute indication* arrives on the dialogue with the commit coordinator, a collision of *ready signals or ready-substitute indication*s has occurred. The node may be the commit coordinator.

If a provider abort occurs on the dialogue with the commit coordinator, the node propagates the result "not-determined" (synchronizing event "continue-unknown" is generated).

State 20.3.3

The node is in the READ-ONLY state. Synchronizing event "enter-read-only-state" or "enter-early-exit-state" has been received.

A C-NOCHANGE confirmation or a C-BEGIN is awaited on the dialogue towards the coordinator. Synchronizing event "Continue-unknown" is awaited on all other dialogues.

If a *ready signal or ready-substitute indication* arrives on the dialogue with the commit coordinator, a collision of *ready signals or ready-substitute indication*s has occurred. The node will never become the commit coordinator.

If a provider abort occurs on the dialogue with the commit coordinator, the node propagates the result "not-determined".

Moreover, this state is used for a node in Early-Exit state. AF-EARLY-EXIT cnf or *rollback indication* is awaited.

State 21.1

This state is valid only for a dialogue with a subordinate. If the node was in READY state and this is not the dialogue towards the coordinator, a commit request has been issued and a *commit confirm* is awaited. If this is the dialogue towards the coordinator, a report on data is awaited (either two-phase or one-phase procedures).

State 21.2

This state is valid only for a dialogue with a subordinate. A *commit confirm* or a report on data is awaited. The next branch will be rolled back if a C-COMMIT confirm or an AF-REPORT (commitRC) indication or AF-REPORT (dataRI) is received.

NOTE – An AF-ABORT (commitRC) indication is received only if the Unchained Transactions functional unit is selected.

State 21.3

This state is valid only for a dialogue with a subordinate. A *commit confirm* has been received or was not awaited. Synchronizing event "Complete-commit" is awaited. If this is the dialogue towards the commit coordinator and a commit indication has been received, commit confirm will be sent with "complete-commit" processing.

State 21.4

This state is valid only for a dialogue with a subordinate that is chaining. Either a *commit confirm* has been received and Rollback has been initiated on this branch or this is the static one-phase chaining subordinate dialogue and rollback is to be initiated. Synchronizing event "Complete-commit" is awaited.

State 21.5

This state is valid only for the dialogue with the superior. This is the dialogue towards the commit coordinator. Synchronizing event "Complete-commit" is awaited.

State 21.5.1

This state is valid only for the dialogue with the superior. Reporting on data applies on the dialogue and synchronizing event "send-report" is awaited.

State 21.5.2

This state is valid only for the dialogue with the superior. The dynamic and the unchained functional units are selected on the dialogue. Commit confirm is awaited.

State 21.5.3

This state is valid only for the dialogue with the superior. Either this is the chaining dialogue with coordination level "one-phase" or the node has issued AF-EARLY-EXIT request and the early-exit was accepted by the superior and in both cases C-BEGIN indication is awaited for completion of commitment.

State 21.5.4

This state is valid only for the dialogue with the superior. Synchronizing event "Complete-commit" is awaited.

State 21.6

This state is valid only for the dialogue with the superior that is chaining. This is the dialogue towards the commit coordinator. The next branch will be rolled back. Synchronizing event "Complete-commit" is awaited if the association has not aborted.

State 21.6.1

This state is valid only for the dialogue with the superior that is chaining. Reporting on data applies on the dialogue and synchronizing event "send-report" is awaited. The next branch will be rolled back.

State 21.6.3

This state is valid only for the dialogue with the superior that is chaining. C-BEGIN indication is awaited. The next branch will be rolled back.

State 21.6.4

This state is valid only for the dialogue with the superior. Synchronizing event "Complete-commit" is awaited. The next branch will be rolled back.

State 23.1

This state is valid only for a dialogue with a subordinate. A *rollback request* has been issued. A rollback confirm is awaited.

State 23.2

This state is valid only for a dialogue with a subordinate. A *rollback indication* or a *rollback confirm* has been received or is not awaited. Synchronizing events "Report-rollback" or "Complete-rollback" are awaited.

State 23.3

This state is valid only for the dialogue with the superior. A TP-ROLLBACK request, Internal event "Rollback by TPPM", or synchronizing event "Rollback-all" has been received. Synchronizing event "Report-rollback" is awaited.

State 23.4

This state is valid only for the dialogue with the superior. A *rollback indication* has been received. Synchronizing event "Report-rollback" is awaited.

State 23.5

This state is valid only for the dialogue with the superior. A rollback report has been issued to the superior. A *rollback confirm* or synchronizing event "Complete-rollback" is awaited.

State 23.6

This state is valid only for the dialogue with the superior. A rollback report has been issued to the superior, and confirmation has been received. The dialogue with the superior *is available for the next transaction*. A C-BEGIN indication or synchronizing event "Complete-rollback" is awaited.

State 23.7

This state is valid only for the dialogue with the superior. A rollback report has been issued to the superior, and confirmation has been received. The dialogue with the superior *is not available for the next transaction*. A TP-DONE request or synchronizing event "Complete-rollback" is awaited.

State 23.8

This state is valid only for the dialogue with the superior. The dialogue with the superior has failed and the conditions for reporting rollback have not been fulfilled. The synchronizing event "Complete-rollback" is awaited.

State 25

This state is valid for a dialogue with a subordinate. This dialogue, having a coordination level of "commitment" or "one-phase", has been terminated during the active state of the transaction without causing a rollback. This "zombie" dialogue will participate in the termination of the transaction like a read-only subordinate.

State 26.1

This state is valid for the node related finite state machine. The NFSM is dormant. No transaction identifier is in use at the node.

State 26.2

This state is valid for the node related finite state machine. The NFSM is awake. A transaction identifier is in use at the node.

State 26.3

This state is valid for the node related finite state machine. The NFSM is awake. The node is in a DECIDED state and rollback is not initiated at the node.

State 26.4

This state is valid for the node related finite state machine. The NFSM is awake. Rollback is initiated at the node.

State 99

On the dialogue with the commit coordinator, the TPPM is in the READY state. A CAF-PLEASE request has been issued. A CAF-GIVE indication is awaited. If a CAF-RECOVER (ready) indication is received, a rollback is triggered.

On all other dialogues, the TPPM is in the DECIDED (commit) state. A CAF-PLEASE request has been issued. A CAF-GIVE indication is awaited.

A.4.1.2 CPM states

State 1

Idle state. No channel exists.

State 2

The channel is free and may be allocated to a TPPM. For a one-way recovery channel, the channel was initiated by this CPM and the AF-BEGIN-DIALOGUE (accepted) confirm has been received. For a two-way-recovery channel, the token is owned (unless the token will arrive as part of the channel establishment procedures managed by the SACF).

State 3

The channel is not free and must not be allocated to a TPPM. For a one-way-recovery channel, the channel was not initiated by this CPM. For a two-way-recovery channel, the token is not owned (and is not expected to arrive as part of the channel establishment procedures managed by the SACF).

State 4

The channel is temporarily owned by a TPPM.

State 5

The channel is established in the two-way recovery mode and the token is awaited to perform recovery.

State 6

A channel is being established. An AF-BEGIN-DIALOGUE confirm is awaited.

State 7

The channel has been detached by the TPPM while C-RECOVER (ready) request was outstanding.

A.4.2 MACF variables

A.4.2.1 Overview

Six categories of variables are defined for MACF:

a) variables that pertain to a dialogue. These variables are created at dialogue establishment time, and are destroyed at termination time of the dialogue or the transaction branch, whichever occurs later. Dialogue variables are prefixed by the letter "D". They are listed in Table A.1;

b) variables pertaining to a channel. These variables are created at channel establishment time, and are destroyed at channel termination time. Channel variables are prefixed by the letter "C". They are listed in Table A.2;

c) variables that pertain to a node. These variables are created at establishment time of the first dialogue that includes the node as part of the dialogue tree and are destroyed at termination time of the node’s last dialogue or the node’s last transaction branch, whichever occurs later. Node variables are prefixed by the letter "N". They are listed in Table A.3;

d) variables that model open system data. System variables are prefixed by the letter "S". They are listed in Table A.4;

e) variables that model a decision local to the node, when there is a choice for the TPPM. Local decision variables are prefixed by the letter "L". Local decision variables reflect a local decision made at the time of reference of the value of the variable. Therefore, the values of these variables are determined newly each time they are referenced. Local decision variables are listed in Table A.5; and

f) variables that are owned by SACF and shared with MACF when MACF is attached to the association. These variables are listed in Table A.6.

Table A.1/X.862 – Dialogue variables

Name	Meaning
D2exp	**two**-phase **exp**ected branch
D2pc	dialogue with **two**-**p**hase **c**ommitment procedures
Da	dialogue establishment **a**ccepted
Dah	dialogue establishment **a**ccepted and **h**eld
Danu	**a**tomic-action identifier **n**ot **u**sed
Danyb	**any** a**b**ort received
Db	dialogue a**b**orted and not available
Dbcr	C-**B**EGIN **c**onfirm **r**eceived
Dbegdi	AF-**BEG**IN-**D**IALOGUE **i**ndication
Dbpart	a**b**ort issued to/received from **part**ner
Dbrid	current **br**anch **id**entifier
Dbridn	**br**anch **id**entifier for **n**ext transaction
Dc	**c**ontrol
Dcancfu	**canc**el **f**unction **u**nit
Dcancr	C-**CANC**EL ind **r**eceived
Dcdfu	**c**ompletion **d**iagnostics **f**unctional **u**nit
Dch	**ch**aining dialogue with a subordinate
Dchat	**ch**annel **at**tached
Dcoor	dialogue towards the commitment **coor**dinator
Dcr	**c**onfirmation **r**equested
Dcrpa	**c**ompletion **r**eport **p**arameter **a**llowed with next TP-DONE req
Dd	TP-**D**ONE request owed
Ddef	AF-**DEF**ER to be sent on prepare
Ddp	**d**ata **p**ermitted (polarized control mode)
Ddyn	**Dyn**amic commit functional unit
De	defer **e**nd-dialogue
Dec	**e**xclusive **c**ontrol of the dialogue
Deefu	**e**arly-**e**xit **f**unctional **u**nit
Deei	AF-**E**ARLY-**EX**IT **i**nd received on the dialogue
Denb	**n**um**b**er of outstanding TP-U-**E**RROR requests
Denbb	**n**um**b**er of outstanding TP-U-**E**RROR requests **b**efore TP-BEGIN-TRANSACTION request
Depnb	AF-U-**E**RROR res**p**onses **n**um**b**er
Dex	**ex**clusive branch
Dfdone	**f**irst TP-**DONE** request received
Dg	defer **g**rant-control
Dgrp	**group** identification for the dialogue
Dh	**H**andshake functional unit
Dhrsfu	**heuristic containment required f**unctional **u**nit
Dimpl	**impl**icit prepare functional unit
Dl	coordination **l**evel
Dnchra	**n**ot **c**ompensatable **h**euristic **r**eport **a**waited
Do	C**o**mmit functional unit
Dopfu	**o**ne-**p**hase **f**unctional **u**nit
Dopi	C-NOCHANGE (result-requested) ind received
Dps	**p**repare semantic **s**ent
Drbrep	**r**oll**b**ack **rep**orted to superior
Drdyi	C-**R**EA**DY** **i**nd received (and no C-NOCHANGE req issued)
Drofu	**r**ead-**o**nly **f**unctional **u**nit
Droi	C-NOCHANGE (result-not-required) ind received
Drrec	*ready signal* or *ready-substitute indication* is **rec**eivable
Drsen	**r**eady signal or *ready-substitute request* is **sen**dable
Drvyp	**r**eco**v**er**y** **p**ending
Dsh	**Sh**ared Control functional unit
Dsopex	**s**tatic **o**ne-**p**hase **ex**clusive branch
Dsup	dialogue with **sup**erior
Dtb	a**b**ort received from **T**PSUI
Du	**U**nchained Transactions functional unit
Dx	transaction e**x**tended

Table A.2/X.862 – Channel variables

Name	Meaning
Caaid	**a**tomic-**a**ction **id**entifier
Cbrid	atomic-action-**br**anch **id**entifier
Cinit	channel **init**iator
Csup	channel to **sup**erior
Ctokr	**tok**en **r**equested

Table A.3/X.862 – Node variables

Name	Meaning
N2exp	**Two**-phase **exp**ected branch at the node
Naaid	current **a**tomic-**a**ction **id**entifier
Naaidn	**a**tomic-**a**ction **id**entifier for **n**ext transaction
Nbrid	superior **br**anch **id**entifier
Nbridn	superior **br**anch **id**entifier for **n**ext transaction
Ncc	early **c**ommit **c**onfirm issued
Nch	**ch**aining dialogue with the superior
Nclw	**c**ommit **l**og **w**ritten
Ncmtr	TP-**C**O**M**MI**T** req **r**eceived
Ncnt	**c**ou**nt** of events
Ncnt2exp	**C**ou**nt** **two**-phase **exp**ected branches
Ncntc	**c**ou**nt** of **c**ompletion events
Ncntcdfu	**c**ou**nt**er for dialogues with completion diagnostics functional unit selected
Ncntcr	**c**ou**nt**er for outstanding **c**ompletion **r**eports from subordinates (only used if Nrpdcr = TRUE)
Ncntge	count of global events
Ncnthr	**c**ou**nt**er for outstanding **h**euristic **r**eports from subordinates (only used if Nrpdhr = TRUE)
Ncntopfu	**c**ou**nt**er for coordinated dialogues with **o**ne-**p**hase **f**unctional **u**nit selected
Ncntrdy	**c**ou**nt**er for awaited **ready** signals and *ready-substitute indications*
Ncr	**c**onfirmation **r**equested with superior
Ncrsev	**c**ompletion **r**eporting – value of the **sev**erity parameter
Ncrst	(subordinate) **c**ompletion **r**eport **st**atus known at the node (Ncntcr = 0)
Ncrud	**c**ompletion **r**eporting – value of the **u**ser-**d**ata parameter
Neer	TP-**E**ARLY-**E**XIT req **r**eceived
Nex	**ex**clusive branch at the node
Nfa	**f**ailure **a**ctions allowed
Nfrb	**f**irst **r**oll**b**ack request
Nhrst	(subordinate) **h**euristic **r**eporting **st**atus known at the node (Ncnthr = 0)
Ni	**i**ntermediate node
Nimpl	**impl**icit prepare functional unit on the dialogue with the superior
Nlf	**leaf** node
Nopi	counter for received C/AF-NOCHANGE (result-requested) ind
Nopr	TP-**O**NE-**P**HASE req **r**eceived
Np	**p**repare indication received
Nr	**r**oot node
Nrdyi	counter for received C-**READY** **i**nd
Nresult	**result** of the transaction (commit, not-determined, no-change)
Nrn	**r**eject **n**ot allowed
Nror	TP-**R**EAD-**O**NLY req **r**eceived
Nrpdcr	**rep**orting on **d**ata to superior applies for **c**ompletion **r**eport and report is not yet sendable (reports from subordinates or TPSUI outstanding)
Nrpdhr	**rep**orting on **d**ata to superior applies for **h**euristic **r**eport and report is not yet sendable (reports from subordinates or TPSUI outstanding)

Table A.3/X.862 – Node variables *(fin)*

Name	Meaning
Nrpend	**r**ollback **pend**ing
Nsopex	**S**tatic **o**ne-**p**hase **ex**clusive branch
Nsubnb	**sub**ordinate **n**um**b**er
Nt	transaction **t**ermination
Ntbicr	**T**PSUI **b**elieves **i**n **c**ompletion **r**eporting
Ntch	**t**ree **ch**ecking at the node
Ntpsui	**TPSUI** created

Table A.4/X.862 – System variables

Name	Meaning
SldD	**l**og-**d**amage **d**ata
SlhD	**l**og-**h**euristic **d**ata
SnD	**n**ode **d**ata

Table A.5/X.862 – MACF local decisions variables

Name	Meaning
Ldbd	**d**ecision that there are **b**ound **d**ata at the TPPM
Ldcanc	**d**ecision to issue C-**CANC**EL req on the dialogue
Lddef	**d**ecision to delay AF-**DEF**ER request
Ldfail	**l**ocal **fail**ure in writing a log record
Ldhrcomp	**d**ecision to **comp**ensate a **h**euristic **r**eport
Ldperm	**d**ecision of **perm**anent failure
Ldprep	**d**ecision to send **prep**are
Ldready	**d**ecision to send a **ready** signal
Ldrej	**d**ecision to **rej**ect the dialogue
Ldretry	**d**ecision to **retry**
Ldretryo	**d**ecision to **retry** on the old channel
Ldt	**d**ecision to **t**erminate the channel
Ldtch	**d**ecision to perform **t**ree extension **ch**ecks by the TPPM
Ldtwr	**d**ecision to have **t**wo-**w**ay-**r**ecovery
Ldunk	**d**ecision to reject, recipient **unk**nown

Table A.6/X.862 – Shared variables with SACF

Name	Meaning
Aaet	peer **AE**-**t**itle
Arrh	**r**eceived **r**ecovery-context-**h**andle
Atokx	**tok**en e**x**pected
Atppm	attached to a **TPPM**
Atwr	**t**wo-**w**ay-**r**ecovery

A.4.2.2 Definitions of MACF types

A.4.2.2.1 Log-record types

Certain MACF variables contain sets of data which are structured into record types. These types are defined here. They are referred to only in A.4.2.3 which defines the MACF variables. Each type definition may be used as a set. Therefore, the type definition specifies the field(s) that uniquely identify the member of the set.

When a variable refers to a type definition, a field in the variable is referenced by the name of the variable, the value of the identifier field(s) in parenthesis (if the type is used as a set), a period, and the name of the field in the type definition.

NOTE 1 – For example, suppose sbbr is a set of Tbranch. To reference the rch field for a branch with the brid of Dbrid, specify "sbbr (Dbrid).rch".

Tbranch (**branch**): Tbranch is a record which contains all necessary information about a single transaction branch (which could be to the superior or a subordinate). This record contains the following fields:

– brid: contains the branch identifier for the specified branch;

– aet: contains the peer AE-title;

– rch: contains the peer partner recovery-context-handle for the branch, if provided;

– readyk: indicates whether a C-READY-RI has flown on this branch or not and its direction. Possible values are (none, ready-received, ready-sent);

  NOTE 2 – "readyk" stands for "kind of ready".

– hreport: indicates whether heuristic reporting is required or not. Possible values are (none, required).

The brid field identifies the record.

Tnode (**node record**): Tnode is a record which contains all of the information required for a node record. Tnode applies to any type of node record. This record contains the following fields:

– aaid: contains the atomic action identifier;

– spbr: contains a Tbranch record for the superior;

– type: contains the type of log record. Possible values are "heuristic-hazard", "heuristic-mixed", "heuristic-initial", "heuristic-final", "log-commit", "log-ready", or NULL;

– sbbr: contains a set of Tbranch records, one for each subordinate.

The aaid and spbr.brid fields identify the record.

The Tnode-records contains the Tbranch-records with readyk = "ready-sent" or with readyk = "ready-received" as soon as their type becomes "log-ready" or "log-commit", i.e. the other Tbranches are removed when a Tnode-record is brought to secure storage.

NOTE 3 – A log-record belongs to the commit coordinator if and only if:

– there is no Tbranch-record with readyk = "ready-sent" and

– the type = "log-commit" (Tnode level).

A.4.2.2.2 Dialogue types

The classification of transaction branches related to tree checking is given in ITU-T Rec. X.861 | ISO/IEC 10026-2 (see definition of *exclusive branch*, *static one-phase exclusive branch* and *two-phase-expected branch*). Most of the relevant information for control of ready-flow is related to the dialogue and given at dialogue establishment time. There is exactly one exception: the check-ready-directions parameter, which occurs only at transaction branch establishment time.

The type of the dialogue identifies the properties of the branches, which can be established on the dialogue.

The following types of dialogues are defined for the dialogue with a subordinate to characterize the possible branches on such a dialogue:

1) a static one-phase branch only if there is tree checking at the node;

2) a static one-phase branch with and without tree checking at the node;

3) an *exclusive branch* only if there is tree checking at the node;

4) an *exclusive branch* with and without tree checking at the node;

5) a *two-phase expected branch* only if there is tree checking at the node;

6) a *two-phase expected branch* only if there is tree checking at the node and the check-ready-directions parameter is set to true;

7) this is a type 8 dialogue if the check-ready-directions parameter is set to false and a type 6 dialogue, otherwise;

8) neither a *two-phase expected branch* nor an *exclusive branch* nor a static one-phase branch, and establishment of a branch is only allowed if there is no tree checking at the node;

9) a *two-phase expected branch* with and without tree checking at the node;

10) a *two-phase expected branch* only if the check-ready-directions parameter is set to true;

11) this is a type 12 dialogue if the check-ready-directions parameter is set to false and a type 10 dialogue, otherwise;

12) a *two-phase expected branch* and establishment of a branch is only allowed if there is no tree checking at the node;

13) establishment of a branch is not allowed on this dialogue (always none-level).

The following types of dialogues are defined for the dialogue with the superior to characterize the possible branches on such a dialogue:

1) an *exclusive branch* only if there is tree checking at the node;

2) this is a type 1 dialogue if the check-ready-directions parameter is set to false and a type 3 dialogue, otherwise;

3) an *exclusive branch* and tree checking is ordered by the superior;

4) neither a *two-phase expected branch* nor an *exclusive branch* nor a static one-phase branch;

5) establishment of a branch is not allowed on this dialogue (always none-level).

The dialogue types are derived from the combinations of functional units and other parameters of TP-BEGIN-DIALOGUE req. These combinations are shown in Table A.7.

Examples:

1) The first dialogue type of group 5 in Table A.7 (branch with a subordinate) characterizes a branch with uptree ready-flow. Either a C-READY indication or C-NOCHANGE (result-not-required) ind is awaited on the branch. This branch is a two-phase-expected branch only if there is tree-checking at the node.

2) The dialogue type of group 12 in Table A.8 (branch with a subordinate) characterizes a branch with up- and downtree ready-flow. Only C-READY request/indication is allowed on the branch. This branch is a two-phase expected branch only if there is no tree-checking at the node and this branch is not allowed if there is tree-checking at the node; the check-ready-directions-parameter is set to false and therefore the subordinate is not ordered to check the tree-extension rules.

Table A.7/X.862 – Parameters and dialogue types for the dialogue with a subordinate

Description of the possible transaction branch	Parameters of the dialogue/branch								Without tree check at the node	With tree check at the node	Resulting group
	co	dyn	op	ro	sp	sb	uct	crd			
static one-phase	ns	na	S	S	na	na	na	na	–	sopex	1
static one-phase	ns	na	S	ns	na	na	na	na	sopex	sopex	2
dynamic 2PC with 1PC	S	S	S	S	t	f	na	na	–	excl	3
dynamic 2PC without 1PC	S	S	ns	S	t	f	na	na	–	excl	3
dynamic 2PC without 1PC	S	S	ns	ns	t	f	na	na	excl	excl	4
dynamic 2PC with 1PC	S	S	S	ns	t	f	na	na	excl	excl	4
static 2PC	S	ns	na	S	na	na	na	na	–	2exp	5
dynamic 2PC without 1PC	S	S	ns	S	f	t	na	na	–	2exp	5
dynamic 2PC without 1PC	S	S	ns	S	t	t	ns	na	–	2exp	5
dynamic 2PC with 1PC	S	S	S	ns	f	t	na	na	–	2exp	5
dynamic 2PC with 1PC	S	S	S	ns	t	t	ns	na	–	2exp	5
dynamic 2PC with 1PC	S	S	S	S	f	t	na	na	–	2exp	5
dynamic 2PC with 1PC	S	S	S	S	t	t	ns	na	–	2exp	5
dynamic 2PC without 1PC	S	S	ns	S	t	t	s	t	–	2exp	6
dynamic 2PC with 1PC	S	S	S	ns	t	t	s	t	–	2exp	6
dynamic 2PC with 1PC	S	S	S	S	t	t	s	t	–	2exp	6
dynamic 2PC with 1PC	S	S	S	S	t	t	s	t/f	–	2exp / na	7
dynamic 2PC without 1PC	S	S	ns	S	t	t	s	f	–	na	8
dynamic 2PC with 1PC	S	S	S	ns	t	t	s	f	–	na	8
dynamic 2PC with 1PC	S	S	S	S	t	t	s	f	–	na	8
static 2PC	S	ns	na	ns	na	na	na	na	2exp	2exp	9
dynamic 2PC without 1PC	S	S	ns	ns	f	t	na	na	2exp	2exp	9
dynamic 2PC without 1PC	S	S	ns	ns	t	t	ns	na	2exp	2exp	9
dynamic 2PC without 1PC	S	S	ns	ns	t	t	s	t	2exp	2exp	10
dynamic 2PC without 1PC	S	S	ns	ns	t	t	s	t/f	2exp	2exp / na	11
dynamic 2PC without 1PC	S	S	ns	ns	t	t	s	f	2exp	na	12
none-level	ns	ns	ns	na	na	na	na	na	–	–	13

Table A.8/X.862 – Parameters and dialogue types for the dialogue with the superior

Description of the possible transaction branch	Parameters of the dialogue/branch								Without tree check at the node	With tree check at the node	Resulting group
	co	dyn	op	ro	sp	sb	uct	crd			
dynamic 2PC without 1PC	S	S	ns	ns	t	t	s	f	–	excl	1
dynamic 2PC without 1PC	S	S	ns	S	t	t	s	f	–	excl	1
dynamic 2PC with 1PC	S	S	S	ns	t	t	s	f	–	excl	1
dynamic 2PC with 1PC	S	S	S	S	t	t	s	f	–	excl	1
dynamic 2PC with 1PC	S	S	ns	ns	t	t	s	t/f	– / na	excl	2
dynamic 2PC with 1PC	S	S	ns	S	t	t	s	t/f	– / na	excl	2
dynamic 2PC with 1PC	S	S	S	ns	t	t	s	t/f	– / na	excl	2
dynamic 2PC with 1PC	S	S	S	S	t	t	s	t/f	– / na	excl	2
static 2PC	S	ns	na	ns	na	na	na	na	na	excl	3
static 2PC	S	ns	na	S	na	na	na	na	na	excl	3
dynamic 2PC without 1PC	S	S	ns	ns	f	t	na	na	na	excl	3
dynamic 2PC without 1PC	S	S	ns	ns	t	t	na	na	na	excl	3
dynamic 2PC without 1PC	S	S	ns	ns	t	t	s	t	na	excl	3
dynamic 2PC without 1PC	S	S	ns	S	f	t	na	na	na	excl	3
dynamic 2PC without 1PC	S	S	ns	S	t	t	ns	na	na	excl	3
dynamic 2PC without 1PC	S	S	ns	S	t	t	s	t	na	excl	3
dynamic 2PC with 1PC	S	S	S	ns	f	t	na	na	na	excl	3
dynamic 2PC with 1PC	S	S	S	ns	t	t	ns	na	na	excl	3
dynamic 2PC with 1PC	S	S	S	ns	t	t	s	t	na	excl	3
dynamic 2PC with 1PC	S	S	S	S	f	t	na	na	na	excl	3
dynamic 2PC with 1PC	S	S	S	S	t	t	ns	na	na	excl	3
dynamic 2PC with 1PC	S	S	S	S	t	t	s	t	na	excl	3
static one-phase	ns	na	S	ns	na	na	na	na	–	–	4
static one-phase	ns	na	S	S	na	na	na	na	–	–	4
dynamic 2PC without 1PC	S	S	ns	ns	t	f	na	na	–	–	4

Table A.8/X.862 – Parameters and dialogue types for the dialogue with the superior *(fin)*

Description of the possible transaction branch	Parameters of the dialogue/branch								Without tree check at the node	With tree check at the node	Resulting group
	co	dyn	op	ro	sp	sb	uct	crd			
dynamic 2PC without 1PC	S	S	ns	S	t	f	na	na	–	–	4
dynamic 2PC with 1PC	S	S	S	ns	t	f	na	na	–	–	4
dynamic 2PC with 1PC	S	S	S	S	t	f	na	na	–	–	4
none-level	ns	ns	ns	ns	na	na	na	na	–	–	5

co	Commit functional unit
dyn	Dynamic commit functional unit
op	One-phase functional unit
ro	Read-only or early-exit functional unit
sp	Superior-may-send-ready parameter
sb	Subordinate-may-send-ready parameter
uc	Unchecked tree functional unit
crd	Check-ready-directions parameter
S	Selected
ns	Not selected
t	Parameter set to 'TRUE'
f	Parameter set to 'FALSE'
na	Not applicable
excl	*Exclusive branch*
2exp	Two-phase-expected branch
sopex	Static-one-phase *exclusive branch*
–	A branch is neither "excl" nor "2exp" nor "sopex"

A.4.2.3 Definitions of MACF variables

The following variables are defined for the MACF state table. MACF variables are Boolean variables, unless otherwise specified.

Caaid (**a**tomic-**a**ction **id**entifier on **c**hannel): the value of Caaid indicates the atomic-action identifier used for recovery of the branch on the channel.

Cbrid (atomic-action-**br**anch **id**entifier on **c**hannel): the value of Cbrid indicates the atomic-action-branch identifier used for recovery of the branch on the channel.

Cinit (channel **init**iator): set to TRUE when the channel is initiated by the CPM.

Csup (channel to **sup**erior): set to TRUE when a channel is requested for recovery to the superior.

Ctokr (**tok**en **r**equested): set to TRUE when the token for the channel is requested by the CPM.

D2exp (**two**-phase **exp**ected branch): when set to TRUE, D2exp indicates that this is a two-phase expected branch.

D2pc (dialogue with **two**-**p**hase **c**ommitment procedures): the variable is used in ready state and decided (commit) state, when set to TRUE, D2pc indicates that a *ready signal* has been received or has been issued.

Da (dialogue establishment **a**ccepted): when related to a dialogue with the superior and set to TRUE, Da indicates that a first request or response has been issued to the superior. When related to a dialogue with a subordinate and set to TRUE, Da indicates that a first indication or confirm has been received.

Dah (dialogue establishment **a**ccepted and **h**eld): when set to TRUE, Dah indicates that a TP-BEGIN-DIALOGUE (accepted) response has been issued in the DECIDED (rollback) state, but its propagation has not occurred yet, because a TP-DONE request is awaited.

Danu (**a**tomic-action identifier **n**ot **u**sed): when set to TRUE, indicates that the atomic-action identifier issued in a C-COMMIT+C-BEGIN request/indication (or an AF-NOCHANGE (result-requested) indication on a chaining static one-phase superior dialogue) has not been used/is not used. When related to a chaining dialogue with a subordinate, indicates that an AF-ABORT (user, commitRC) indication or an AF-ABORT-AND-REPORT (commitRC) indication [or an AF-ABORT (user, nochangeRC) indication or an AF-ABORT-AND-REPORT (user, nochangeRC) indication] has been received from the subordinate. When related to a chaining dialogue with a superior, indicates that the TPSUI issued a TP-U-ABORT request for the dialogue in the READY- or ONE-PHASE-state, and that an AF-ABORT (user, commitRC) request or an AF-ABORT-AND-REPORT (commitRC) request [or an AF-ABORT (user, nochangeRC) request or an AF-ABORT-AND-REPORT (user, nochangeRC) request] has been issued or is to be issued to the superior TPPM in response to a C-COMMIT+C-BEGIN indication [or C-NOCHANGE (result-requested) indication].

Danyb (**any** a**b**ort received): when set to TRUE, Danyb indicates that the dialogue has been or will be detached, or that the SAO is no longer attached (an SAF-ASSOCIATION-LOST indication has been received, or an SAF-DETACH-ASSOCIATION request has been issued). Danyb is true when Db, Dbpart, or Dtb is true.

Db (dialogue a**b**orted and not available): when set to TRUE, Db indicates that the dialogue has been detached. Db is true when an SAF-DETACH-ASSOCIATION request has been issued or an SAF-ASSOCIATION-LOST indication has been received. Only one of Db, Dtb or Dbpart may be true.

Dbcr (C-**B**EGIN **c**onfirm **r**eceived): when set to TRUE, Dbcr indicates that a C-BEGIN confirm has been received. Dbcr is used to check the validity of an AF-END-DIALOGUE indication and an AF-ABORT indication when the Unchained Transactions functional unit is selected.

Dbegdi (AF-**BEG**IN-**D**IALOGUE **i**ndication): Dbegdi is used to save the AF-BEGIN-DIALOGUE indication so that the parameters from this indication are available when a TP-BEGIN-DIALOGUE indication is issued after the C-BEGIN indication arrives.

Dbpart (a**b**ort issued to/received from **part**ner): when set to TRUE, Dbpart indicates that an AF-ABORT (user) request/indication or an AF-ABORT-AND-REPORT request/indication has been issued to or received from the partner TPPM. Only one of Db, Dtb or Dbpart may be true. In the case of commitment, Dbpart is used to detect protocol errors. In the case of rollback, Dbpart is used to repeat the abort if necessary in the event of a rollback collision.

Dbrid (current **br**anch **id**entifier): the value of Dbrid indicates the atomic-action-branch identifier to a subordinate for the current transaction.

Dbridn (**br**anch **id**entifier for **n**ext transaction): the value of Dbridn indicates the atomic-action-branch identifier for the next transaction.

Dc (**c**ontrol): when set to TRUE, Dc indicates that the TPSUI had control at the beginning of the transaction branch. Dc denotes which TPSUI will acquire the control of the dialogue, upon completion of rollback, should rollback occur.

Dcancfu (**canc**el **f**unction **u**nit): when set to TRUE, Dcancfu indicates that the cancel functional unit is selected.

Dcancr (C-**CANC**EL ind **r**eceived): when set to TRUE, Dcancr indicates that C-CANCEL indication has been received and that no C-CANCEL request is to be issued on the dialogue.

Dcdfu (**c**ompletion **d**iagnostics **f**unctional **u**nit): when set to TRUE, Dcdfu indicates that the completion diagnostics functional unit is selected.

Dch (**ch**aining dialogue with a subordinate): when set to TRUE, Dch indicates that the dialogue *is chaining*.

Dchat (**ch**annel **at**tached): when set to TRUE, Dchat indicates that a channel is attached to the TPPM for recovery of a particular branch.

Dcoor (dialogue towards the commitment **coor**dinator): the value is 'TRUE' if this is the dialogue to the commit coordinator. Dcoor is set to "TRUE" if a *ready signal or ready-substitute request* has been issued. If a collision of *ready signals or ready-substitute indications* follows, Dcoor may be set back to 'FALSE', if the node is the commit coordinator.

Dcr (**c**onfirmation **r**equested): when set to TRUE, Dcr denotes on a dialogue with a subordinate that the TPPM has received a TP-BEGIN-DIALOGUE (*confirmation* = "always") request and has not yet received an AF-BEGIN-DIALOGUE confirm.

Dcrpa (**c**ompletion **r**eport **p**arameter **a**llowed with next TP-DONE request): when set to TRUE, Dcrpa indicates that completion report paramaters are allowed with the next TP-DONE request. Completion report parameters are allowed if the completion diagnostics functional unit is selected on the dialogue with the superior and moreover only after TP-COMMIT request/TP-COMMIT indication and after TP-COMPLETION-REPORT indication if there was no provider abort of the superior dialogue.

NOTE 1 – The provider abort may not be known by the TPSUI (provider abort after TP-U-ABORT request), see Ntbicr.

Dd (TP-**D**ONE request owed): when set to TRUE, Dd indicates that a *TP-DONE request is owed*. When set to FALSE, Dd indicates that a *TP-DONE request is not owed* by the TPSUI.

Ddef (AF-**DEF**ER to be sent on prepare): when set to TRUE, an AF-DEFER request is to be issued when the AF-PREPARE request is issued. The type of the AF-DEFER request is determined by the values of De and Dg. When set to FALSE, no AF-DEFER request is issued when the AF-PREPARE request is issued.

Ddp (**d**ata **p**ermitted): when set to TRUE and if the Polarized Control functional unit is selected, Ddp indicates that a TP-DATA indication may be received by the superior TPSUI, after it has issued a TP-PREPARE request.

Ddyn (**dyn**amic commit functional unit): when set to TRUE, Ddyn indicates that the dynamic commit functional unit is selected.

De (defer **e**nd-dialogue): when set to TRUE, De indicates that either an AF-DEFER (end-dialogue) request has been issued or an AF-DEFER (end-dialogue) indication has been received.

Dec (exclusive **c**ontrol of the dialogue): when set to TRUE, Dec indicates that the Polarized Control functional unit is selected on the dialogue and the dialogue state machine has control of the dialogue.

Deefu (**e**arly-**e**xit **f**unctional **u**nit): when set to TRUE, Deefu indicates that the early-exit functional unit is selected.

Deei (AF-**E**ARLY-**E**XIT **i**nd received on the dialogue): when set to TRUE, Deei indicates that AF-EARLY-EXIT ind has been received and AF-EARLY-EXIT response has been issued (as an immediate response).

Denb (**n**um**b**er of outstanding TP-U-**E**RROR requests): Denb is a variable of type Integer, used in shared control mode only. Denb indicates the number of outstanding TP-U-ERROR requests. Denb is incremented by 1 upon issuance of a TP-U-ERROR request. Denb is decremented by 1 upon receipt of an AF-U-ERROR confirm, AF-HANDSHAKE indication, or AF-END-DIALOGUE (*confirmation* = TRUE) indication. Denb is set to zero upon occurrence of a rollback.

Denbb (**n**um**b**er of outstanding TP-U-**E**RROR requests **b**efore TP-BEGIN-TRANSACTION request): Denbb is a variable of type Integer, used in shared control mode and Unchained Transactions only.

Denbb indicates the number of TP-U-ERROR requests that were outstanding when the TP-BEGIN-TRANSACTION was received. Denbb is set to the value of Denb when a TP-BEGIN-TRANSACTION request is received. Denbb is decremented whenever Denb is decremented.

Depnb (AF-U-**E**RROR res**p**onse **n**um**b**er): Depnb is a variable of type Integer, used in shared control mode only. Depnb indicates the number of TP-U-ERROR responses that shall be issued after an AF-BEGIN-DIALOGUE response is issued.

Dex (**ex**clusive branch): when set to TRUE, Dex indicates that this is an *exclusive branch.*

Dfdone (**f**irst TP-**DONE** request received): when set to TRUE, Dfdone indicates that the first TP-DONE request after TP-COMMIT indication or after a rollback initiating indication has been received. When set to FALSE, Dfdone indicates that a TP-DONE request with a heuristic-report parameter may be received, subject to the value of Dd.

Dg (defer **g**rant-control): when set to TRUE, Dg indicates that either an AF-DEFER (grant-control) request has been issued or an AF-DEFER (grant-control) indication has been received.

Dgrp (**gr**ou**p** identification for the dialogue): the variable is used to identify the functional units and other parameters of a specific dialogue which are related to tree checking. This variable is set during dialogue establishment and branch creation in the appropriate subcell(s). The variable is used to calculate the local state of a transaction tree during establishment of a branch on the dialogue using TP-BEGIN-DIALOGUE request, TP-BEGIN-TRANSACTION request, AF-BEGIN-DIALOGUE indication, C-BEGIN indication and AF-BEGIN-TRANSACTION indication and during completion of a transaction. See A.4.2.2 for the description of the values.

Dh (**H**andshake functional unit): when set to TRUE, Dh indicates that the Handshake functional unit is selected.

Dhrsfu (heuristic containment required functional unit): when set to TRUE, Dhrsfu indicates that the heuristic containment required functional unit is selected (the variable name was derived from alternative name for the functional unit used in a draft).

Dimpl (**impl**icit prepare functional unit): when set to TRUE, Dimpl indicates that the implicit prepare functional unit is selected.

Dl (coordination **l**evel): Dl reflects the value of the coordination level. When set to TRUE, Dl indicates that the coordination level is "commitment" or "one-phase"; when set to FALSE, Dl indicates that the coordination level is "none".

Dnchra (**n**ot **c**ompensatable **h**euristic **r**eport **a**waited): when set to TRUE, Dnchra indicates that a heuristic report is awaited on the dialogue (Dhrsfu is set to FALSE) and that the reporting to the superior (if necessary) is not possible until the report is available on a dialogue with two-phase procedures or with one-phase procedures or the report is no longer receivable (on a dialogue with one-phase procedures). If a non-empty heuristic report arrives on a dialogue with Dnchra set to FALSE, the report does not affect the value of the heuristic record at the node.

Do (c**o**mmit functional unit): when set to TRUE, Do indicates that the commit functional unit is selected.

Dopfu (**o**ne-**p**hase **f**unctional **u**nit): when set to TRUE, Dopfu indicates that the one-phase functional unit is selected.

Dopi [C-NOCHANGE (result-requested) indication received]: a C-NOCHANGE (result-requested) indication was received on the dialogue.

NOTE 2 – Only one of Drdyi, Dopi, Droi may be 'TRUE' for a single branch.

Dps (**p**repare semantic **s**ent): Dps is set to TRUE after an AF-PREPARE request or a *ready signal* or *ready-substitute request* is issued.

Drbrep (**r**oll**b**ack **rep**orted to superior): when set to TRUE, Drbrep indicates that rollback has been reported to the superior. Drbrep is used by all transaction branches to avoid resetting Ncnt once rollback has been reported to the superior and a TP-DONE request becomes owed.

Drdyi [C-**READY** **i**ndication received (and no C-NOCHANGE request issued)]: a C-READY indication was received on the dialogue and there was no forced collision initiated by the node or after sending a C-READY indication, the node's *ready signal* is no longer valid (collision with C-READY-RI issued by the neighbour and the node is the coordinator).

NOTE 3 – Only one of Drdyi, Dopi, Droi may be 'TRUE' for a single branch.

Drofu (**r**ead-**o**nly **f**unctional **u**nit): when set to TRUE, Drofu indicates that the read-only functional unit is selected.

Droi [C-NOCHANGE (result-not-required) indication received]: a C-NOCHANGE (immediate) indication or C-NOCHANGE (result-not-required) indication has been received on the dialogue.

NOTE 4 – Only one of Drdyi, Dopi, Droi may be 'TRUE' for a single branch.

Drrec [***r**eady signal* or AF/C-NOCHANGE (result-requested) indication is **rec**eivable]: when set to TRUE, Drrec indicates that a ready signal *or* AF/C-NOCHANGE (result-requested) indication is receivable on the dialogue, i.e.:

1) a subordinate dialogue without dynamic commit functional unit and commit functional unit selected;

2) the superior dialogue with coordination level one-phase;

3) a subordinate dialogue with dynamic commit functional unit selected and superior-may-send-ready set to TRUE;

4) the superior dialogue with dynamic commit functional unit selected and subordinate-may-send-ready set to TRUE.

Drsen [***r**eady signal* or AF/C-NOCHANGE (result-requested) request is **sen**dable]: when set to TRUE, Drsen indicates that a ready signal or AF/C-NOCHANGE (result-requested) request is sendable on the dialogue, i.e.:

1) the superior dialogue without dynamic commit functional unit and commit functional unit selected;

2) a subordinate dialogue with coordination level one-phase;

3) the superior dialogue with dynamic commit functional unit selected and subordinate-may-send-ready set to TRUE;

4) a subordinate dialogue with dynamic commit functional unit selected and superior-may-send-ready set to TRUE.

Drvyp (**r**eco**v**er**y** **p**ending): when set to TRUE, Drvyp enables the occurrence of "Retry-recovery" for that branch.

Dsh (**Sh**ared Control functional unit): when set to TRUE, Dsh indicates that the Shared Control functional unit is selected. When set to FALSE, Dsh indicates that the Polarized Control functional unit is selected.

Dsopex (**s**tatic **o**ne-**p**hase **ex**clusive branch): when set to TRUE, Dsopex indicates that this is a static one-phase *exclusive branch*.

Dsup (dialogue with **sup**erior): when set to TRUE, Dsup indicates that the dialogue or transaction branch is with the superior.

Dtb (a**b**ort received from **T**PSUI): when set to TRUE, Dtb indicates that the TPSUI has issued a TP-U-ABORT request, but the issuance of the AF-Service has not yet occurred. Once the AF-ABORT request has been issued or an AF-ABORT indication is received, Dtb is set to FALSE, and Db or Dbpart is set as appropriate. Only one of Db, Dtb or Dbpart may be true.

Du (**U**nchained Transactions functional unit): when set to TRUE, Du indicates that the Unchained Transactions functional unit is selected.

Dx (transaction e**x**tended): Dx is a variable of type Boolean which is set to TRUE when a TP-BEGIN-TRANSACTION request is received and is set to FALSE when a C-BEGIN confirm is received. Dx is used to determine if an AF-END-DIALOGUE indication or an AF-ABORT (user, dataRI) indication is valid for a subordinate dialogue with a coordination level "none".

Ldbd (**d**ecision that there are **b**ound **d**ata at the TPPM): the local condition is 'TRUE' if there are bound data at the TPPM and the bound data were accessed during the current transaction. The local condition is 'FALSE', if there are no bound data at the TPPM or the bound data were not accessed during the current transaction.

Ldcanc (**d**ecision to issue C-**CANC**EL request on the dialogue): when set to TRUE, a C-CANCEL request is issued on the dialogue. This local decision is only used if the cancel functional unit is selected on the dialogue and rollback is initiated at the node.

Lddef (**d**ecision to delay AF-**DEF**ER request): when set to TRUE, an AF-DEFER request will be issued when an AF-PREPARE request is issued. When set to FALSE, an AF-DEFER request is issued immediately.

Ldfail (**l**ocal **fail**ure in writing a log record): when set to TRUE, Ldfail indicates that the TPPM cannot write a log record or set the bound data to the proper state if there are any.

Ldhrcomp (**d**ecision to **comp**ensate a **h**euristic **r**eport): when set to TRUE, Ldhrcomp indicates that a heuristic report (awaited or received) should not affect the log-heuristic-record.

Ldperm (**d**ecision of **perm**anent failure): when set to TRUE, Ldperm indicates that the *diagnostic* parameter shall be set to "permanent-failure". When set to FALSE, Ldperm indicates that the *diagnostic* parameter shall be set to "transient-failure".

Ldprep (**d**ecision to send **prep**are): when set to TRUE, Ldprep indicates that a AF-PREPARE request shall be issued when entering the termination phase of the transaction although all other conditions for sending prepare (see GENPREP) are not satisfied.

Ldready (**d**ecision to send a ***ready** signal*): when set to TRUE, Ldready indicates that the TPPM chooses to issue a signal of readiness. When set to FALSE, Ldready indicates that the TPPM is still waiting for the last ready signal or *ready-substitute indication*. The decision is only evaluated if the TPPM is able to receive the last ready signal or *ready-substitute indication* on the dialogue (Drrec is set to TRUE on this dialogue) and if this is no *exclusive branch*.

Ldrej (**d**ecision to **rej**ect the dialogue): when set to TRUE, Ldrej indicates that the TPPM takes a local decision to reject the dialogue establishment.

NOTE 5 – This definition applies to both the initiator side (local reject) and responder side.

Ldretry (**d**ecision to **retry**): when set to TRUE, Ldretry indicates:

a) for a TPPM that it may issue a C-RECOVER (retry-later) response when either the transaction outcome is not yet known to respond to a CAF-RECOVER (ready) indication, or when all commit confirms have not yet been received to respond to a CAF-RECOVER (commit) indication; and

b) for a CPM that it may issue a C-RECOVER (retry-later) response when the value of the recovery-context-handle does not allow it to determine if a TPPM can be found.

Ldretryo (**d**ecision to **retry** on the **o**ld channel): when set to TRUE, Ldretryo indicates that a C-RECOVER (retry-later) response will be sent on the old channel. Since this variable is used in conjunction with Ldretry in cases where at least one C-RECOVER (retry-later) response shall be sent, this variable shall not be set to FALSE if Ldretry is set to FALSE in the same subcell.

Ldt (**d**ecision to **t**erminate the channel): when set to TRUE, Ldt indicates that channel utilization is to be terminated.

Ldtch (local **d**ecision to perform **t**ree extension **ch**ecks by the TPPM): when set to TRUE, there is a local decision to perform tree extension checks by the TPPM. This local decision is evaluated when the node becomes part of a transaction tree, i.e. the first branch is established at the node, without being ordered by the superior in the transaction tree to check the tree extension rules.

Ldtwr (**d**ecision to have **t**wo-**w**ay-**r**ecovery): when set to TRUE, Ldtwr indicates that the channel shall be established in the two-way recovery mode. When set to FALSE, Ldtwr indicates that the channel shall be established in the one-way recovery mode.

Ldunk (**d**ecision to reject, recipient **unk**nown): when set to TRUE, Ldunk indicates that the dialogue is to be rejected because an association could not be established and the diagnostic parameter to be issued on the TP-BEGIN-DIALOGUE confirm is to be "recipient-unknown". When set to FALSE, the dialogue is rejected for other reasons and the diagnostic parameter of the TP-BEGIN-DIALOGUE confirm is "no-reason-given".

N2exp (**Two**-phase **exp**ected branch at the node): when set to TRUE, N2exp indicates that a two-phase expected branch exists at the node, i.e. Ncnt2exp > 0.

Naaid (current-**a**tomic **a**ction **id**entifier): the value of Naaid indicates the atomic action identifier for the current transaction.

Naaidn (**a**tomic-**a**ction **id**entifier for **n**ext transaction): the value of Naaidn indicates the atomic action identifier for the next transaction.

Nbrid (superior **br**anch **id**entifier): the value of Nbrid indicates the atomic-action-branch identifier to the superior.

Nbridn (superior **br**anch **id**entifier for **n**ext transaction): the value of Nbridn indicates the atomic-action-branch identifier to the superior for the next transaction.

Ncc (early **c**ommit **c**onfirm issued): when set to TRUE, Ncc indicates that a commit confirm has been issued on the dialogue towards the coordinator prior completion of the termination phase.

Nch (**ch**aining dialogue with the superior): when set to TRUE, Nch indicates that the dialogue with the superior *is chaining.*

Nclw (**c**ommit **l**og **w**ritten): when set to TRUE, Nclw indicates that the log commit record has been written at the node and early commit confirmation might be possible now.

Ncmtr (TP-**COMM**IT request **r**eceived): when set to TRUE, Ncmtr indicates that a TP-COMMIT request has been received from the TPSUI.

Ncnt (**cou**n**t** of events): the number of events that must occur before the next node state transition can occur. Ncnt is decremented for each relevant event [see A.3 rule f), counting mechanism].

Ncnt2exp (**Cou**n**t** **two**-phase **exp**ected branches): number of two-phase expected branches at the node.

Ncntc (**cou**n**t** of **c**ompletion events): the number of events that must occur before commitment/rollback is complete for a node and the one-time completion actions can be performed. Ncntc is decremented for each dialogue tidying during termination.

Ncntcdfu (**coun**t**er for dialogues with **c**ompletion **d**iagnostics **fu**nctional unit selected): diagnostic parameters with TP-ROLLBACK indication are only allowed if there is at least one coordinated dialogue with completion diagnostics functional unit at the node.

Ncntcr [**coun**t**er for outstanding **c**ompletion **r**eports from subordinates (only used if Nrpdcr = TRUE)]: The number of completion reports needed from subordinates for reporting of completion status **(on data)** to the superior. The counter is only used if completion report is necessary on the superior dialogue, i.e. completion report functional unit is selected and there was no provider abort or protocol error on the superior dialogue. Ncntcr is decremented for each completion report arriving at the node and if a completion report is no longer receivable on a subordinate dialogue. If Ncntcr is decremented to 0, then Ncrst is set to TRUE. If there is only completion reporting and reporting is still possible on the dialogue with the superior and there are no completion report parameters allowed with the next TP-DONE request (Ntbicr) and Ncntcr = 0 (Ncrst = TRUE), then the synchronizing event "send-report" is generated.

Ncntge (**cou**n**t** of **g**lobal **e**vents): number of state-machines having processed the *transaction completion request* (TP-COMMIT request, TP-ONE-PHASE request, TP-READ-ONLY request).

Ncnthr [**coun**t**er for outstanding **h**euristic **r**eports from subordinates (only used if Nrpdhr = TRUE)]: The number of heuristic reports needed from subordinates for reporting of heuristic status **(on data)** to the superior. The counter is only used if heuristic report is necessary on the superior dialogue, i.e. heuristic containment required functional unit is not selected and either one-phase procedures are used on the dialogue with the superior and there was no provider abort or protocol error on the superior dialogue or two-phase procedures are used on the dialogue with the superior. Ncnthr is decremented for each not-compensatable heuristic report arriving at the node and if a heuristic report is no longer receivable on a subordinate dialogue (one-phase procedures). If Ncnthr is decremented to 0, then Nhrst is set to TRUE. If there is only heuristic reporting and reporting is still possible on the dialogue with the superior and the first TP-DONE request is not owed and Ncnthr = 0 (Nhrst = TRUE), then the synchronizing event "send-report" is generated.

Ncntopfu (count**er for coordinated dialogues with **o**ne-**p**hase **f**unctional **u**nit selected): this counter is used to check the service rule for TP-ONE-PHASE request (at least one coordinated dialogue with the one-phase functional unit selected).

Ncntrdy (**coun**t**er for awaited **r**e**ady** signals or *ready-substitute indications*): number of expected *ready signals* or *ready-substitute indications*.

Ncr (**c**onfirmation **r**equested with the superior): when set to TRUE, Ncr indicates that a dialogue establishment indication is outstanding on the dialogue with the superior and therefore that a TP-BEGIN-DIALOGUE response is awaited.

Ncrsev (**c**ompletion **r**eporting – value of the **sev**erity parameter): the severity parameter of the last TP-DONE request for completion reporting.

Ncrst [(subordinate) **c**ompletion **r**eport **st**atus known at the node (Ncntcr = 0)]: when set to TRUE, Ncrst indicates that there is no completion report outstanding from a subordinate. Ncrst is only used if there is completion reporting on data to the superior.

Ncrud (**c**ompletion **r**eporting – value of the **u**ser**-d**ata parameter): the completion-data parameter of the last TP-DONE request for completion reporting to the superior (if possible).

Neer (TP-**E**ARLY-**E**XIT request **r**eceived): when set to TRUE, Neer indicates that a TP-EARLY-EXIT request has been received from the TPSUI.

Nex (**ex**clusive branch at the node): when set to TRUE, Nex indicates that there is an *exclusive branch* at the node.

Nfa (**f**ailure **a**ctions allowed): when set to TRUE, Nfa indicates that TP-U-ABORT requests are authorized during transaction termination.

Nfrb (**f**irst **r**oll**b**ack): used to determine that a TP-ROLLBACK request or TPPM initiated rollback request is processed so specific actions may be taken only once. Set to TRUE when a TP-ROLLBACK request or TPPM initiated rollback request is received by the first branch state machine.

Nhrst [(subordinate) **h**euristic **r**eporting **st**atus known at the node (Ncnthr = 0)]: when set to TRUE, Nhrst indicates that there is no not-compensatable heuristic report outstanding from a subordinate. Nhrst is only used if there is heuristic reporting on data to the superior. Nhrst being TRUE does not include a statement about the TPSUI's contribution to the heuristic status.

Ni (**i**ntermediate node): when set to TRUE, Ni indicates that the node is an intermediate node of the transaction tree.

Nimpl (**impl**icit prepare functional unit on the dialogue with the superior): if set to TRUE, Nimpl indicates that the implicit prepare functional unit is selected on the dialogue with the superior.

Nlf (**leaf** node): when set to TRUE, Nlf indicates that the node is a leaf node of the transaction tree.

Nopi [counter for received C/AF-NOCHANGE (result-requested) indication]: number of C/AF-NOCHANGE (result-requested) indication at the node.

Nopr (TP-**O**NE-**P**HASE request **r**eceived): when set to TRUE, Nopr indicates that a TP-ONE-PHASE request has been received from the TPSUI.

Np (**p**repare indication received): when set to TRUE, Np means that an AF-PREPARE indication or *ready signal* or *ready substitute indication* is received from the superior or that the implicit prepare functional unit is selected on the dialogue with the superior.

Nr (**r**oot node): when set to TRUE, Nr indicates that the node is the root node of the transaction tree.

Nrdyi (counter for received C-**REA**D**Y** **i**ndication): number of *ready signals* (C-READY indication) at the node.

Nresult [**result** of the transaction (commit, not-determined, no-change)]: this variable is used to carry the result-value of a C-NOCHANGE-RC. The possible values are "commit", "not-determined" and "no-change". For these results, commitment procedures are used.

NOTE 6 – A "rollback" result is not used as a value for Nresult. If the outcome of the transaction is rollback, the rollback-procedures are used.

Nrn (**r**eject **n**ot allowed): when set to TRUE, Nrn denotes that the TPSUI has issued a request or response on any dialogue, and thus indicates that the dialogue with the superior can no longer be rejected.

Nror (TP-**R**EAD-**O**NLY request **r**eceived): when set to TRUE, Nror indicates that a TP-READ-ONLY req has been received from the TPSUI.

Nrpdcr [**rep**orting on **d**ata to superior applies for **c**ompletion **r**eport and report is not yet sendable (reports from subordinates or TPSUI outstanding)]: when set to TRUE, Nrpdcr indicates that completion reporting applies (completion diagnostics functional unit selected and no provider abort) and that separated reporting is necessary on the dialogue with the superior but reporting is not yet possible (there are outstanding reports). The node is on the occ path and not the root and Nresult = commit.

Nrpdhr [**rep**orting on **d**ata to superior applies for **h**euristic **r**eport and report is not yet sendable (reports from subordinates or TPSUI outstanding)]: when set to TRUE, Nrpdhr indicates that heuristic reporting applies and that separated reporting is required on the dialogue (or the channel) with the superior but reporting is not yet possible (there are outstanding reports). The node is on the occ path and not the root and Nresult = commit.

Nrpend (**r**ollback **pend**ing): when set to TRUE, Nrpend indicates that the synchronizing event "Rollback-next-trans" has been received.

Nsopex (**s**tatic **o**ne-**p**hase **ex**clusive branch): when set to TRUE, Nsopex indicates that a *static one-phase exclusive branch* exists at the node.

Nsubnb (**sub**ordinate **n**um**b**er): Nsubnb is a variable of type Integer. Nsubnb indicates the number of subordinates.

Nt (transaction **t**ermination): when set to TRUE, Nt indicates that the transaction branch has entered the termination phase, and that the transaction tree can no longer grow.

Ntbicr (TPSUI **b**elieves **i**n **c**ompletion **r**eporting): when set to TRUE, Ntbicr indicates that completion report parameters are allowed with the next TP-DONE request if, in addition, since entering termination phase or since the last TP-DONE request a TP-COMPLETION-REPORT indication has been received. Completion report parameters are allowed if the completion diagnostics functional unit is selected on the dialogue with the superior and moreover only after TP-COMMIT request /TP-COMMIT indication and after TP-COMPLETION-REPORT indication if – to the knowledge of the TPSUI – there was no provider abort of the superior dialogue.

NOTE 7 – The provider abort may not be known by the TPSUI (provider abort after TP-U-ABORT request).

Ntch (**t**ree **ch**ecking at the node): this variable, when set to TRUE, indicates that there is *tree checking* at this node. When the node enters a transaction tree, the variable is set to TRUE either upon a local decision to check tree extension rules or after being ordered by the superior node in the transaction tree to check the tree extension rules. The variable is set (from TRUE) to FALSE if there is no branch for a next transaction after completion of the current transaction. If Ntch is set to TRUE, this is done during the first invocation of procedure TREESET. Therefore, during establishment of the first branch in Table A.13, Ntch is still 'FALSE'.

Ntpsui (**TPSUI** created): when set to TRUE, Ntpsui indicates that the TPSUI corresponding to the TPPM has been created.

SldD (**l**og-**d**amage **d**ata): SldD is a variable of type Set of Tnode. SldD represents the set of log-damage records that are kept by an open system for appropriate heuristic reporting.

NOTE 8 – To reference the value of the type component of the log record for a member of this set with an "aaid" value of Naaid and an "spbr" value of Nbrid, specify "SldD (Naaid, Nbrid).type".

SlhD (**l**og-**h**euristic **d**ata): SlhD is a variable of type Set of Tnode. SlhD represents the set of log-heuristic records that are kept by an open system for appropriate heuristic reporting.

NOTE 9 – To reference the value of the type component of the log record for a member of this set with an "aaid" value of Naaid and an "spbr" value of Nbrid, specify "SlhD (Naaid, Nbrid).type".

SnD (**n**ode **d**ata): SnD is a variable of type Set of Tnode. SnD represents the system data attached to transactions that have been initiated and that are not yet complete.

NOTE 10 – To reference the value of the type component of the log record for a member of this set with an "aaid" value of Naaid and an "spbr" value of Nbrid, specify "SnD (Naaid, Nbrid).type". To reference the value of the "aet" field for a subordinate branch identified by Dbrid, specify "Snd (Naaid, Nbrid).sbbr (Dbrid).aet".

All elements of SldD, SlhD, and SnD whose type component is non-NULL shall remain present in the variables after a node crash, and all elements whose type component is NULL shall be absent from the variables after a node crash.

A.4.2.4 Initialization of MACF variables

MACF variables are initialized as follows:

Danyb and Db: initialized to TRUE after a transaction node crash.

All other variables are initialized as follows:

a) variables of type Boolean are initialized to FALSE;

b) variables of type Integer are initialized to zero; and

c) variables of type Octet string are initialized to EMPTY.

The initial state of the NFSM is given in A.2.2.4.

A.4.3 MACF events

A.4.3.1 Internal events

The following internal events, defined for the MACF state table, occur within the scope of a single branch:

Delay-recovery

Abbreviation for "Delay Recovery" as defined in 11.4.1.

Internal error

Abbreviation for "Internal Error", as defined in 11.3.21, 11.3.22, and 11.3.23.

Rewrite-log

Abbreviation for "Rewriting intermediate record" as defined in 11.4.9.

Protocol error

Abbreviation for "Protocol Error", as defined in 11.3.21, 11.3.22, and 11.3.23.

Retry-recovery

Abbreviation for "Retry recovery", as defined in 11.4.4.

Terminate-channel

Abbreviation for "Terminating a Channel", as defined in 11.4.6.

The following internal events, defined for the MACF state table, occur on all branches of a TPPM of a transaction tree:

Heuristic-damage-comp

Abbreviation for "Heuristic damage compensation for subtree", as defined in 11.4.2.

Heuristic-decision

Abbreviation for "Taking a Heuristic Decision" as defined in 11.4.5. Heuristic decision carries one parameter, called "heuristic-report", which can take the value of either "heuristic-final", "heuristic-initial", "heuristic-hazard", or "heuristic-mix".

Restart-TPPM

Abbreviation for "TPPM creation after node crash" as defined in 11.4.7.

Rollback-by-TPPM

Abbreviation for "TPPM initiated rollback" as defined in 11.4.8. Moreover, this internal event is used to trigger Rollback at the node after a Collision of C-READY indications, when the collision winner is not able to write the log commit record.

A.4.3.2 Synchronizing events

The following synchronizing events are defined for the MACF state table:

Activate-nfsm

This event is generated when the node becomes part of a transaction tree. The NFSM enters the awake state.

Complete-commit

This event is generated when commitment is completed at a node. When the dialogue towards the commit coordinator receives this, it issues a *commit response* if a commit indication has been received. All other state machines complete the transaction. C-BEGIN request is sent on any dialogue with a subordinate that is in a chain of transactions.

Complete-rollback

This event is generated when rollback is complete and the next transaction may begin (if the Chained Transactions functional unit is selected on any dialogue). C-BEGIN request is sent on any dialogue with a subordinate that is available for the next transaction. Other branches involved in the rollback transaction are deleted from the transaction tree.

Continue-commit

"Continue-commit" is used by each dialogue (except the dialogue towards the commit coordinator) to issue a *commit request* or a C-NOCHANGE response. The result "commit" is known at the node.

Continue-unknown

"Continue-unknown" is used by each dialogue (except the dialogue towards the commit coordinator) to issue a C-NOCHANGE (not-determined) response. The result is unknown and will stay unknown at the node.

Cr-allowed

This event is generated by a dialogue on when a TP-COMPLETION-REPORT indication has been issued and the TPSUI believes in completion reporting (see Ntbicr). Each dialogue sets Dcrpa to TRUE.

Cr-not-allowed

This event is generated by the superior dialogue in the termination phase, when a TP-P-ABORT request is issued. The TPSUI no longer believes in completion reporting. Each dialogue sets Dcrpa to FALSE.

Deactivate-nfsm

This event is generated when the node is no longer part of a transaction tree. The NFSM enters the dormant state.

Enter-ready-state

This event is used by the dialogue towards the commit coordinator to issue a C-READY request.

Enter-one-phase-state

This event is used by the dialogue towards the commit coordinator to issue a C-NOCHANGE (result-requested) request or an AF-NOCHANGE (result-requested) request.

Enter-read-only-state

This event is used by the dialogue towards the superior to issue a C-NOCHANGE (result-not-required) request.

Enter-early-exit-state

This event is used by the dialogue towards the superior to issue an AF-EARLY-EXIT request.

Log-rewritten

This event is generated when writing of a log-commit record was successful at a node having received a *commit indication* (see procedure 11.4.9 and action REWRLOG).

One-ready

This event is generated by the state machine receiving the last *ready signal or ready-substitute indication* or *transaction completion request* and is used if there is only one *ready signal* at the node and there are no bound data at the node. The state machine which has received the C-READY indication sends a C-NOCHANGE indication (if otherwise possible) or makes a decision.

Report-rollback

This event is generated when all the conditions required to report the rollback to the superior are fulfilled. The dialogue to the superior uses this event to issue a *rollback response* or *rollback request* to the superior.

Report-Status

This event is used [when the DECIDED (commit) state is entered] to collect the reporting status at the node if there is reporting on data on the dialogue with the superior. It is generated by the dialogue with the superior and used by dialogues with subordinates (Ncntcr, Ncnthr, Nhrst, Ncrst are modified if a report is awaited on the dialogue and this report is necessary for reporting to the superior).

Rollback-all

This event is generated when a rollback occurs at a node. It is used by dialogues with subordinates to issue a *rollback request*.

Rollback-next-trans

This event is generated by a subordinate during commitment upon either receipt of A(-P)-ABORT indication, TP-U-ABORT request when a C-COMMIT+C-BEGIN indication has previously been received. All dialogues with subordinates must issue a *rollback request* if the *commit confirm* has been received.

Send-prepare

This event is generated when the node enters the termination phase (not rollback termination) and no *ready signal* or *ready-substitute indication* is sendable.

Send-report

This event is generated if the report status is known at the node and used by the state machine towards the superior to send a report.

Send-ready?

This event is generated by the state machine upon receiving a *ready signal or ready-substitute indication* or *transaction completion request* if one *ready signal or ready-substitute indication* is missing at the node. The event is used by the state machine which has not received a *ready signal or ready-substitute indication*. A *ready signal* or *ready-substitute request* may be sent by this state machine or rollback is initiated or there is a local decision to wait for the last *ready signal* or *ready-substitute indication* (Ldready).

Set-done-true

Indicates to all dialogues that a *TP-DONE request is* now *owed*. Each dialogue sets Dd to TRUE.

A.4.4 MACF actions

A.4.4.1 Functions

Variables which contain sets are manipulated by the functions described below. These functions are used to add and delete members of the set and to determine if a given member of a set exists.

addBranch (variable, brid, aet)

Adds a new member of a set of type Tbranch. The variable parameter specifies a variable (or field of a variable) which is a Set of Tbranch. The brid parameter specifies the value of the "brid" field of Tbranch which identifies the branch in the set. The aet parameter specifies the value of the "aet" field in the new member.

addNode (variable, aaid, spbrid)

Adds a new member of a set of type Tnode. The variable parameter specifies a variable (or field of a variable) which is a Set of Tnode. The aaid parameter specifies the value of the "aaid" field of Tnode, and the value of spbrid specifies the value of the "spbr.brid" field of Tnode. The aaid and the spbrid parameters identify the node record. The new record is created with its type component initialized to NULL and its sbbr component initialized to the empty set.

delBranch (variable, brid)

Deletes a member of a set of type Tbranch. The variable parameter specifies a variable (or field of a variable) which is a Set of Tbranch. The brid parameter specifies the value of the "brid" field of Tbranch which identifies the branch in the set.

delNode (variable, aaid, spbrid)

Deletes a member of a set of type Tnode. The variable parameter specifies a variable (or field of a variable) which is a Set of Tnode. The aaid parameter specifies the value of the "aaid" field of Tnode, and the value of spbrid specifies the value of the "spbr.brid" field of Tnode. The aaid and the spbrid parameters identify the node record.

memsb (variable, aaid, sbbrid)

Determines if a node record in the specified variable exists which refers to the specified *atomic-action-identifier* and subordinate *atomic-action-branch-identifier*. The aaid parameter specifies the value of the "aaid" field of Tnode which identifies the node record in the set. The sbbrid parameter specifies the value of a "sbbr.brid" field which identifies a branch within the node record. If the specified record is found, TRUE is returned; otherwise, FALSE is returned.

memsp (variable, aaid, spbrid)

Determines if a node record in the specified variable exists which refers to the specified *atomic-action-identifier* and superior *atomic-action-branch-identifier*. The aaid parameter specifies the value of the "aaid" field of Tnode which identifies the node record in the set. The spbrid parameter specifies the value of a "spbr.brid" field which identifies the branch with the superior within the node record. If the specified record is found, TRUE is returned; otherwise, FALSE is returned.

A.4.4.2 Actions on services

Table A.9 lists actions that issue service primitives. These actions are named according to the following convention:

First character

A	AF-
C	C- or CAF-
P	P-
S	SAF-
T	TP-
U	U-

Next characters

AB	ABORT
AR	ABORT-AND-REPORT
ASE	ASE
BD	BEGIN-DIALOGUE
BE	BEGIN
BT	BEGIN-TRANSACTION
CR	COMPLETION-REPORT
DE	DEFERRED (end-dialogue)
DET	DETACH
DG	DEFERRED (grant-control)

DT	DATA
ED	END-DIALOGUE
EE	EARLY-EXIT
GC	GRANT-CONTROL
GIV	GIVE
HR	HEURISTIC-REPORT
HS	HANDSHAKE
HSGC	HANDSHAKE-AND-GRANT-CONTROL
NC	NOCHANGE
OP	ONE-PHASE
PAB	P-ABORT
PL	PLEASE
PR	PREPARE
R	REPORT
RB	ROLLBACK
RC	REQUEST-CONTROL
RE	RECOVER
RO	READ-ONLY
RY	READY
TOKG	TOKEN-GIVE
TOKP	TOKEN-PLEASE
UAB	U-ABORT
UE	U-ERROR

Service Primitive type

rq	Request
i	Indication
rs	Response
c	Confirm

Miscellaneous parameter values

A	result = **A**ccepted
CU	type = **c**lean-**u**p
F	confirmation = **F**alse
NU	type = **n**ot-**u**sed
P	type = **p**rovider
RO	**r**ecovery functional unit selected, channel-utilization = **o**ne-way-recovery
RT	**r**ecovery functional unit selected, channel-utilization = **t**wo-way-recovery
RU	result = **R**ejected (**u**ser)
RP	result = **R**ejected (**p**rovider)
SB	**Su**b**o**rdinate
SP	**Sup**erior
U	type = **u**ser
NC	result = **n**o-**c**hange
ND	result = **n**ot-**d**etermined

UP	diagnostic = **u**ser-**p**rotocol-error
X	Inherited parameter value
TR	diagnostic = begin-**t**ransaction-**r**eject
TWR	reason = **T**wo-**w**ay-**r**ecovery

Mapping parameter values

a	abortRI
d	dataRI
r	rollbackRI
rbc	rollbackRC
rd	recoverDoneRC
c	commitRI
crc	commitRC

Source parameter

SAVE uses the parameters as specified in the service primitive contained in a variable

The remaining characters qualify the action in a manner specific to the service being issued.

NOTE – For example, TUABiR issues a TP-U-ABORT indication with the rollback parameter set to TRUE.

A.4.4.3 Actions on variables

For actions that manipulate MACF variables, the following conventions apply:

First character: V

The name of the variable being set begins at the second character.

The final characters are either:

–	C	(set to commit);
–	DEC	(decrement by one);
–	F	(set to FALSE);
–	INC	(increment by one);
–	SAVE	(save the most recently received or issued specified service primitive in the specified variable);
–	T	(set to TRUE); or
–	ND	(set to not-determined);
–	1..13	(set to 1..13).

NOTE – An example is "VdaT", for "set Da to TRUE". Another example is "VDgrp3", for "set Dgrp to 3".

Table A.9/X.862 – MACF actions on services (*sheet 1 of 4*)

Action name	Parameter settings/service primitive issued
AABrqPa	AF-ABORT (provider, abortRI) req on the dialogue
AABrqPaUP	– Set the *diagnostic* parameter to "user-protocol-error". AF-ABORT (provider, abortRI) req on the dialogue
AABrqPrTR	– Set the *diagnostic* parameter to "begin-transaction-reject". AF-ABORT (provider, rollbackRI) req
AABrqUd	AF-ABORT (user, dataRI) req
AABrqUr	AF-ABORT (user, rollbackRI) req
AABrqUrbc	AF-ABORT (user, rollbackRC) req
ABDrq	AF-BEGIN-DIALOGUE (Dialogue fu selected) req
ABDrqRO	AF-BEGIN-DIALOGUE (Recovery fu selected, one-way-recovery) req
ABDrqRT	AF-BEGIN-DIALOGUE (Recovery fu selected, two-way-recovery) req
ABDrsAd	AF-BEGIN-DIALOGUE (accepted, dataRI) rsp

Table A.9/X.862 – MACF actions on services (*sheet 2 of 4*)

Action name	Parameter settings/service primitive issued
ABDrsArbc	AF-BEGIN-DIALOGUE (accepted, rollbackRC) rsp
ABDrsRPd	AF-BEGIN-DIALOGUE (rejected(provider), dataRI) rsp
ABDrsRUd	AF-BEGIN-DIALOGUE (rejected(user), dataRI) rsp
ABDrsRUr	AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) rsp
ABDrsRUrbc	AF-BEGIN-DIALOGUE (rejected(user), rollbackRC) rsp
ABTrq	AF-BEGIN-TRANSACTION req
ADErq	AF-DEFER (end-dialogue) req
ADGrq	AF-DEFER (grant-control) req
AEDrq	AF-END-DIALOGUE (confirmation = TRUE) req
AEDrqF	AF-END-DIALOGUE (confirmation = FALSE) req
AEDrs	AF-END-DIALOGUE rsp
AEErs	AF-EARLY-EXIT rsp
AGCrq	AF-GRANT-CONTROL req
AHSrq	AF-HANDSHAKE req
AHSrs	AF-HANDSHAKE rsp
AHSGCrq	AF-HANDSHAKE-AND-GRANT-CONTROL req
AHSGCrs	AF-HANDSHAKE-AND-GRANT-CONTROL rsp
APRrq	AF-PREPARE req
ARCrq	AF-REQUEST-CONTROL req
ARrqHrdC	– Set the *atomic-action-identifier* parameter to AAI. – Set the *atomic-action-branch-identifier* parameter to BI. – Set the *heuristic-report* parameter to SldD (AAI, BI).type. AF-REPORT (recoverDoneRC) req
ATOKGrqTWR	AF-TOKEN-GIVE (two-way-recovery) req
ATOKPrq	AF-TOKEN-PLEASE req
AUErq	AF-U-ERROR req
AUErs	AF-U-ERROR rsp
CAFDETrqCU	– Set the *type* parameter to "clean-up". CAF-DETACH req
CAFDETrqF	– Set the *type* parameter to "free". CAF-DETACH req
CAFDETrqNU	– Set the *type* parameter to "not-used". CAF-DETACH req
CAFFAILi	CAF-FAIL ind to the transaction branch identified by the *atomic-action-identifier* value of Caaid and the *atomic-action-branch-identifier* value of Cbrid.
CAFGIVi	CAF-GIVE ind to the transaction branch identified by the *atomic-action-identifier* value of Caaid and the *atomic-action-branch-identifier* value of Cbrid.
CAFPLrqSB	– Set the *AE-title* parameter to SnD (Naaid, Nbrid).sbbr (Dbrid).aet. – Set the *atomic-action-identifier* parameter to Naaid. – Set the *atomic-action-branch-identifier* parameter to Nbrid. – Set the *superior* parameter to FALSE. CAF-PLEASE req
CAFPLrqSP	– Set the *AE-title* parameter to SnD (Naaid, Nbrid).spbr.aet. – Set the *atomic-action-identifier* parameter to Naaid. – Set the *atomic-action-branch-identifier* parameter to Nbrid. – Set the *superior* parameter to TRUE. CAF-PLEASE req
CAFREiC	CAF-RECOVER (commit) ind to the transaction branch identified by the *atomic-action-identifier* value of AAI and the *atomic-action-branch-identifier* value of BI
CAFREiR	CAF-RECOVER (ready) ind to the transaction branch identified by the *atomic-action-identifier* value of AAI and the *atomic-action-branch-identifier* value of BI

Table A.9/X.862 – MACF actions on services (*sheet 3 of 4*)

Action name	Parameter settings/service primitive issued
CBErq	– Set the *atomic-action-identifier* parameter to Naaid. – Set the *atomic-action-branch-identifier* parameter to Dbrid. C-BEGIN req
CNCrsND	C-NOCHANGE (not-determined) rsp
CRBrq	C-ROLLBACK req
CRBrs	C-ROLLBACK rsp
CRErsDC	– Set the *atomic-action-identifier* parameter to AAI. – Set the *atomic-action-branch-identifier* parameter to BI. C-RECOVER (done) rsp
CRErsRTSB	– Set the *atomic-action-identifier* parameter to Naaid. – Set the *atomic-action-branch-identifier* parameter to Dbrid. C-RECOVER (retry-later) rsp
CRErsRTSP	– Set the *atomic-action-identifier* parameter to Naaid. – Set the *atomic-action-branch-identifier* parameter to Nbrid. C-RECOVER (retry-later) rsp
CRErsRTC	– Set the *atomic-action-identifier* parameter to AAI. – Set the *atomic-action-branch-identifier* parameter to BI. C-RECOVER (retry-later) rsp
CRErsU	– Set the *atomic-action-identifier* parameter to Naaid. – Set the *atomic-action-branch-identifier* parameter to Dbrid. C-RECOVER (unknown) rsp
CRErsUC	– Set the *atomic-action-identifier* parameter to AAI. – Set the *atomic-action-branch-identifier* parameter to BI. C-RECOVER (unknown) rsp
CRYrq	C-READY req
SDETrqBF	SAF-DETACH-ASSOCIATION (begin-fear) req
SDETrqCB	SAF-DETACH-ASSOCIATION (begin-indication-expected) req
SDETrqF	SAF-DETACH-ASSOCIATION (free) req
SDETrqRB	– Set the *Retain-queue* parameter to FALSE. SAF-DETACH-ASSOCIATION (rollback-indication-expected) req
SDETrqRBC	– Set the *Retain-queue* parameter to FALSE. SAF-DETACH-ASSOCIATION (rollback-confirm-expected) req
SDETrqRBCR	– Set the *Retain-queue* parameter to TRUE. SAF-DETACH-ASSOCIATION (rollback-confirm-expected) req
SDETrqRBR	– Set the *Retain-queue* parameter to TRUE. SAF-DETACH-ASSOCIATION (rollback-indication-expected) req
TBDcRP	– Set the *rollback* parameter to FALSE. – Set the *diagnostic* parameter to "no-reason-given". TP-BEGIN-DIALOGUE (reject(provider)) cnf
TBDcRPr	– Set the *rollback* parameter to TRUE. – Set the *diagnostic* parameter to "no-reason-given". TP-BEGIN-DIALOGUE (reject(provider)) cnf
TBDcRPru	– Set the *rollback* parameter to TRUE. – Set the *diagnostic* parameter to "recipient-unknown". TP-BEGIN-DIALOGUE (reject(provider)) cnf
TBDcRPu	– Set the *rollback* parameter to FALSE. – Set the *diagnostic* parameter to "recipient-unknown". TP-BEGIN-DIALOGUE (reject(provider)) cnf
TBDcX	– Set the *rollback* parameter to FALSE. TP-BEGIN-DIALOGUE cnf

Table A.9/X.862 – MACF actions on services (*sheet 4 of 4*)

Action name	Parameter settings/service primitive issued
TBDcXr	– Set the *rollback* parameter to TRUE. TP-BEGIN-DIALOGUE cnf
TBDi	TP-BEGIN-DIALOGUE ind
TBDiSAVE	– Set the parameters to those in the AF-BEGIN-DIALOGUE ind saved in Dbegdi. TP-BEGIN-DIALOGUE ind
TBTi	TP-BEGIN-TRANSACTION ind
TDEi	TP-DEFERRED-END-DIALOGUE ind
TDGi	TP-DEFERRED-GRANT-CONTROL ind
TDTi	TP-DATA ind
TEDc	TP-END-DIALOGUE cnf
TEDi	TP-END-DIALOGUE ind
TEEi	TP-EARLY-EXIT ind
TGCi	TP-GRANT-CONTROL ind
THRi	TP-HEURISTIC-REPORT ind
THRiH	– Set the *heuristic-report* parameter = "heuristic-hazard". TP-HEURISTIC-REPORT ind
THSc	TP-HANDSHAKE cnf
THSGCc	TP-HANDSHAKE-AND-GRANT-CONTROL cnf
THSGCi	TP-HANDSHAKE-AND-GRANT-CONTROL ind
THSi	TP-HANDSHAKE ind
TOPi	TP-ONE-PHASE ind
TPABi	– Set the *rollback* parameter to FALSE. TP-P-ABORT ind
TPABiBTED	– Set the *rollback* parameter to FALSE. – Set the *diagnostic* parameter to "begin-transaction-end-dialogue-collision". TP-P-ABORT ind
TPABiBTEDr	– Set the *rollback* parameter to TRUE. – Set the *diagnostic* parameter to "begin-transaction-end-dialogue-collision". TP-P-ABORT ind
TPABiBTR	– Set *diagnostic* parameter to "begin-transaction-reject". – Set the *rollback* parameter to FALSE. TP-P-ABORT ind
TPABiED	– Set the *rollback* parameter to FALSE. – Set the *diagnostic* parameter to "end-dialogue-collision". TP-P-ABORT ind
TPABiR	– Set the *rollback* parameter to TRUE. TP-P-ABORT ind
TPABiUP	– Set the *rollback* parameter to FALSE. – Set the *diagnostic* parameter to "user-protocol-error". TP-P-ABORT ind
TPRi	TP-PREPARE ind
TRCi	TP-REQUEST-CONTROL ind
TROi	TP-READ-ONLY ind
TRYi	TP-READY ind
TUABi	– Set the *rollback* parameter to FALSE. TP-U-ABORT ind
TUABiR	– Set the *rollback* parameter to TRUE. TP-U-ABORT ind
TUEi	TP-U-ERROR ind
UASErq	U-ASE req

A.4.4.4 Actions with free-form names

Table A.10 lists the actions with free formed names.

Table A.10/X.862 – MACF actions with free-form names (*sheet 1 of 3*)

Name	Meaning
ABDET	association **ab**orted or **det**ached
ABDETR	association **ab**orted or **det**ached in **R**EADY state
ABPTNR	**ab**orted dialogue with **partn**e**r**
ABTPSUI	**ab**orted by **TPSUI**
ADDBRSB	**add br**anch **sub**ordinate
ADDBRSP	**add br**anch **sup**erior
BEGTRANS	TP-**BEG**IN-**TRANS**ACTION
CANCEL	issue C-**CANCEL** req
CBEAFTCO	**C-BE**GIN **aft**er **co**mmit/not-determined/early-exit
CBEAFTRB	**C-BE**GIN **aft**er **r**oll**b**ack
CMPCOM	**comp**lete **com**mitment one-time actions
CMPCOMSB	**comp**leting **com**mitment with **sub**ordinate
CMPCOMSP	**comp**leting **com**mitment with **sup**erior
CMPRB	**Comp**lete **r**oll**b**ack
COMREQ	issue **com**mit **req**uest
COMRSP	issue **com**mit **resp**onse
COUNTCOM	**count com**mitment confirm event
COUNTCR	**count c**ompletion **r**eport events
COUNTGE	**count g**lobal **e**vent
COUNTRB	**count r**oll**b**ack confirm event
COUNTRDY	**count r**ea**dy** events
COUNTREP	**count rep**orting status events
COUNTREPDO	**count rep**orting with TP-**DO**NE req
CPSAP	**c**lose the **PSAP**
CRDYRESET	**C-R**EA**DY** indication or substitute received, **reset** the readyk field for the branch
CRDYSET	**C-R**EA**DY** indication received, **set** the readyk field for the branch
CRERTRS	issue **C-RE**COVER**ret**ry-later **r**espon**s**e
CRNALL	**c**ompletion **r**eport parameters **n**ot **all**owed with next TP-DONE req
DECDENB	**Dec**rement **Denb**
DECISION	the node is the commit coordinator
DEFREQ	send AF-**DEF**ER **req**uest
DELBR	**del**ete **br**anch
DELBRANCH	**del**ete **branch** in system variable
DELBRO	**del**ete **b**ranch after **r**ead-**o**nly or early-exit
DELIMIT	**delimit** dialogue
DELNBRANCH	**del**ete **n**ext **branch** in system variable
DIALOGUE	operate on **dialogue**
DISCARDS	**discard s**eparator PDUs

Table A.10/X.862 – MACF actions with free-form names (*sheet 2 of 3*)

Name	Meaning
EARLYC	**early** commit confirmation
GENPREP	**gen**erate AF-**PREP**ARE req with termination request
INITDIASB	**init**ialize **dia**logue with **sub**ordinate
INITDIASP	**init**ialize **di**alogue with **sup**erior
INITMACF	**init**ialize **MACF**
INITRB	**init**iate **r**oll**b**ack
INITREPSP	**init**ialize **rep**orting on the OCC-path, **sup**erior dialogue
LOGDAM	**log dam**age
LOGDAMH	**log dam**age **h**azard
LOGDAMRB	**log dam**age **r**oll**b**ack
LOGHD	**log h**euristic **d**ecision
LOGREMOVE	**log**-heuristic, log-damage **remove**
NOTCHAIN	**not chain**ing
NXTBR	commence **next br**anch in chain
NXTTRAN	**next tran**saction
NEWCHANNEL	operate on **new channel**
OLDCHANNEL	operate on **old channel**
OPSAP	**o**pen the **PSAP**
OWEDONE	**owe** a TP-**DONE** request
OWEDONECO	**owe** a TP-**DONE** request after **co**mmit indication and adjust the count of events expected
PREPREQ	issue AF-**PREP**ARE **req**
RBNEXTSB	**r**oll**b**ack **next** transaction **sub**ordinate
RBREQ	issue **r**oll**b**ack **req**uest
RBRSPNOAB	issue **r**oll**b**ack **resp**onse **no ab**ort
RBRSPAB	issue **r**oll**b**ack **resp**onse with **ab**ort
RECCOM	**rec**eive **com**mit indication
RECCOM-OP	**rec**eive **com**mit order if the node is **o**ne-**p**hase or read-only
RECVRCOMI	issue C-**RECOVER** (**com**m**i**t) request
RECVRCOMR	issue C-**RECOVER** (**com**mit) **r**equest after CAF-RECOVER (ready) ind
RECVRDONE	issue C-**RECOVER** (**done**) response
RECVRRDY	issue C-**RECOVER** (**r**ea**dy**) request
REJTRAN	**rej**ect **tran**saction
RESETAAIDN	**reset** the **a**tomic **a**ction **id**entifier for the **n**ext transaction
RESETD	**reset d**ialogue variables
REWRLOG	**rewr**iting intermediate **log**
SAVEAAIDN	**save** the **a**tomic **a**ction **id**entifier for the **n**ext transaction
SAVECR	**save** the **c**ompletion **r**eport parameters of the TP-DONE req
SEND1PC	**send** AF/C-**NOCHANGE (result-requested) req**
SEND2PC	**send C-READY req**
SENDEE	**send** AF-**E**ARLY-**E**XIT req

Table A.10/X.862 – MACF actions with free-form names (*sheet 3 of 3*)

Name	Meaning
SENDRDY?	**send** ***ready*** *signal/ready substitute indication* **?**
SENDREP?	**send rep**ort to superior if possible – without abort
SENDREP?AB	**send rep**ort to superior if possible – with abort
SENDRO	**send C-NOCHANGE (result-not-required) req**
SETAAID	**set** TPPM **a**tomic-**a**ction atomic-action-branch, and superior **id**entifiers
SETDIAG	**set diag**nostic
SETDIAGBD	**set diag**nostic on AF-**B**EGIN-**D**IALOGUE response
SETDIAGEC	**set diag**nostic – **e**arly-exit-transaction-**c**ompletion-collision
SETDIAGLO	**set diag**nostic – **lo**cal-rollback
SETDIAGSB	**set diag**nostic – **s**u**b**ordinate-rollback
SETDIAGSP	**set diag**nostic – **sup**erior-rollback
SETDIAGTP	**set diag**nostic on **TP**-**P**-ABORT-indication
SETDIAGUC	**set diag**nostic – **u**ser-data-transaction-**c**ompletion-collision
SETTOKX	**set** A**tokx** to TRUE
SNDORDCD	**s**e**nd** a NOCHANGE request **or** make a **d**ecision – forced **c**ollision or **d**ecision
TRBi	issue **T**P-**R**OLL**B**ACK **i**nd
TREERESET	**reset tree** extension variables
TREESET	**set tree** extension variables
TREP	issue **rep**ort indication to **T**PSUI

[ABDET] (association **ab**orted or **det**ached**)**

Invoked when the association has been aborted or detached.

– If Dsup and ^Dtb

If there is no TP-U-ABORT req, the abort of the superior dialogue is known by the TPSUI (calling subcell) and completion reporting is no longer possible (in commit and in rollback-termination):

– set Ntbicr to FALSE.

– Set Db and Danyb to TRUE.

– Set Dtb and Dbpart to FALSE.

Reset ready signal flags and functional units. If the dialogue state machine enters state 25 in active state (zombie), it acts like a read-only subordinate with a defer action after receiving an A-P-ABORT request:

– set Dopfu to FALSE;

– set Drofu to FALSE;

– set Do to FALSE;

– set Droi to FALSE;

– set Dopi to FALSE;

– set Drdyi to FALSE;

– set Deei to FALSE.

[ABDETR] (association **ab**orted or **det**ached in **R**EADY-, ONE-PHASE- and READ-ONLY state)

Invoked when the association has been aborted or detached, but keep ready signal flags for continue-commit processing.

- If Dsup and ^Dtb

  If there is no TP-U-ABORT req, the abort of the superior dialogue is known by the TPSUI (calling subcell) and completion reporting is no longer possible (in commit and in rollback-termination):

  - set Ntbicr to FALSE.
- Set Db and Danyb to TRUE.
- Set Dtb and Dbpart to FALSE.

[ABPTNR] (**ab**orted dialogue with **p**ar**tn**e**r**)

Invoked when notification of a dialogue abort is either received from or issued to the partner TPPM.

- Set Dbpart and Danyb to TRUE;
- set Dtb to FALSE.

[ABTPSUI] (**ab**orted by **TPSUI**)

Invoked upon receipt of a TP-U-ABORT request from TPSUI.

- Set Dtb and Danyb to TRUE.

[ADDBRSB] (**add br**anch **sub**ordinate)

Invoked when a branch with a subordinate is added to a transaction. Adds one for the *ready signal* or *ready-substitute indication* and one for the *transaction completion request*.

- Set Dl to TRUE.
- If ^Nr and ^Ni and ^Nlf:
  - set Nr to TRUE;
  - set Naaid to a new unique value;
  - set Nbrid to NULL;
  - addNode (SnD, Naaid, Nbrid);
  - generate "activate-nfsm";
  - add one unit for the NFSM;
  - set Ncntge to 1;
  - set Ncntrdy to 1.
- If Nlf:
  - set Nlf to FALSE;
  - set Ni to TRUE.
- Add one unit to Ncntge.
- Add one unit to Ncntrdy.
- If Dcdfu:
  - add one unit to Ncntcdfu;
  - add one unit to Nsubnb.
- If ^Dsh:
  - set Dc to TRUE.

- If ^Du:
  - set Dch to TRUE.
- Set Dbrid to a new unique value.
- Add Branch (SnD (Naaid, Nbrid).sbbr, Dbrid, Aaet).
- Set SnD (Naaid, Nbrid).sbbr (Dbrid).rch to Arrh.
- Set SnD (Naaid, Nbrid).sbbr (Dbrid).readyk to none.
- If Dhrsfu:
  - set SnD (Naaid, Nbrid).sbbr (Dbrid).hreport to none.
- If ^Dhrsfu:
  - set SnD (Naaid, Nbrid).sbbr (Dbrid).hreport to required.

[ADDBRSP] (**add br**anch **sup**erior)

Invoked when a transaction is created upon indication from the superior. Sets the counters to account for *transaction completion request* and for the *ready signal* or *ready-substitute indication*.

- Set Dl to TRUE;
- set Nlf to TRUE.

Add one unit for the branch and one unit for the NFSM:

- set Ncntge to 2;
- set Ncntrdy to 2;
- set Naaid to the *atomic-action-identifier* parameter;
- set Nbrid to the *atomic-action-branch-identifier* parameter.

This is always the first transaction branch at the node:

- generate "activate-nfsm";
- addNode (SnD, Naaid, Nbrid);
- set SnD (Naaid, Nbrid).spbr.rch to Arrh;
- set SnD (Naaid, Nbrid).spbr.readyk to none.
- If Dhrsfu:
  - set SnD (Naaid, Nbrid).spbr.hreport to none.
- If ^Dhrsfu:
  - set SnD (Naaid, Nbrid).spbr.hreport to required.
- If ^Du:
  - set Nch to TRUE.
- If Da:
  - issue a C-BEGIN rsp.
- If Dcdfu:
  - set Ntbicr to TRUE;
  - add one unit to Ncntcdfu.

[BEGTRANS] (TP-**BEG**IN-**TRANS**ACTION)

Invoked to flag that a TP-BEGIN-TRANSACTION request has been issued and to record the number of TP-U-ERROR requests that are outstanding at that time.

- Set Denbb to Denb;
- set Dx to TRUE.

[CANCEL] (issue C-**CANCEL** request)

Invoked if rollback is initated at the node (for the current or the next transaction).

- If Dcancfu and ^Dcancr
  - If Ldcanc:
    - issue C-CANCEL req.

[CBEAFTCO] (**C-BE**GIN **aft**er **co**mmit **/** not-determined / early-exit)

Invoked upon receipt of a C-BEGIN indication after receiving commit or not-determined (late begin) on a static one-phase chaining superior dialogue or AF-EARLY-EXIT confirm on a chaining superior dialogue.

NOTE 1 – If the node is read-only and the superior dialogue is chaining, C-BEGIN indication has been received in state 20.3.3 – Nresult = not-determined, if the node is early-exit or one-phase and the superior dialogue is chaining, C-BEGIN indication has been received in state 21.5.3.

- Issue a C-BEGIN rsp.
- If Do

  If this is an early-exit node, the aaid for the next transaction is stored in Naaidn and Naaid is set to the value of Naaidn when the complete action is invoked at the node:

  - delNode (SnD, Naaid, Nbrid);
  - set Naaidn to the *atomic-action-identifier* parameter;
  - set Nbridn to the *atomic-action-branch-identifier* parameter;
  - addNode (SnD, Naaidn, Nbridn);
  - set SnD (Naaidn, Nbridn).spbr.rch to Arrh.
- If ^Do

  NOTE 2 – This is a static one-phase chaining superior dialogue. The atomic action identifier for the next transaction was received with AF-NOCHANGE ind.
- Continue.

[CBEAFTRB] (**C-BE**GIN **aft**er **r**oll**b**ack)

Invoked upon receipt of a C-BEGIN indication.

- Issue a C-BEGIN rsp;
- delNode (SnD, Naaid, Nbrid);
- set Naaid to the *atomic-action-identifier* parameter;
- set Nbrid to the *atomic-action-branch-identifier* parameter;
- addNode (SnD, Naaid, Nbrid);
- set SnD (Naaid, Nbrid).spbr.rch to Arrh.

[CMPCOM] (**comp**lete **com**mitment one-time actions)

Counts an event to complete dialogue tidying at the end of commitment for a single branch. When all branches have been tidied completes the one-time actions associated with local commitment completion reporting for a node. Also handles reporting the rollback of the next transaction if that is occurring.

Recalculate variables for tree extension checks.

- If ^Dch or (Dsup and ^Nch)

  This branch is not in the transaction tree for the next transaction:

  - invoke [TREERESET].

- Subtract one unit from Ncntc.
- If Ncntc = 0

    Perform one-time actions at end of commitment:

    - set Naaid to Naaidn;
    - set Nbrid to Nbridn.
    - If ^Nr and ^Ni and ^Nlf

        The node is no longer in a transaction tree:

        - set Ntch to FALSE.
    - Set Naaidn to NULL.
    - Set Nbridn to NULL.
    - If Nresult = commit:
        - issue TP-COMMIT-COMPLETE ind.
    - If Nresult = not-determined:
        - issue TP-UNKNOWN-COMPLETE ind.
    - If ^Nrpend:
        - set Nt to FALSE.

        Activate NFSM if there is a branch for the next transaction, otherwise deactivate NFSM.

        - If Nr or Ni or Nlf:
            - generate "activate-nfsm".
        - If ^Nr and ^Ni and ^Nlf:
            - generate "deactivate-nfsm".
    - If Nrpend:
        - set the *diagnostic* parameter to "rollback-was-pending";
        - issue a TP-ROLLBACK ind;
        - set Nt and Nfa to TRUE;
        - set Nrpend to FALSE;
        - generate "Set-done-true".

        Reset transaction completion request:

        - set Ncmtr to FALSE;
        - set Nopr to FALSE;
        - set Nror to FALSE;
        - set Neer to FALSE.

[CMPCOMSB] (**c**o**mp**leting **com**mitment with **s**u**b**ordinate)

Completes the commitment processing for a single branch with a subordinate.

- If ^Dsh
    - If Dg:
        - set Dc to FALSE;
        - set Dec to FALSE.
    - If ^Dg:
        - set Dc to TRUE;
        - set Dec to TRUE.

- If Dch and ^D2pc

  Chaining dialogue with subordinate – late begin (this is the chaining two-phase subordinate part from COMREQ).

  - If Naaidn = NULL

    Commence a new Atomic-action:

    - set Naaidn to a new unique value;
    - set Nbridn to a new unique value;
    - addNode (SnD, Naaidn, Nbridn).
  - Set Dbridn to a new unique value.
  - AddBranch (SnD (Naaidn, Nbridn).sbbr, Dbridn, Aaet).
  - Set SnD (Naaidn, Nbridn).sbbr (Dbridn).rch to Arrh.
  - Issue C-BEGIN req.
- If Dcoor and D2pc

  Issue commit confirm, dynamic and unchained f.u. selected.

  - If Dtb and ^Db and ^Ncc:
    - issue an AF-ABORT (commitRC) req;
    - set Dbpart to TRUE;
    - set Dtb to FALSE.
  - If ^Dtb and ^Db and ^Ncc:
    - issue a C-COMMIT rsp.
  - If Db and Dchat and ^Ncc:
    - invoke [RCVRDONE];
    - issue CAF-DETACH (free) req;
    - set Dchat to FALSE.
  - Set Dbrid to Dbridn.
  - Set Dbridn to NULL.
  - If ^Db:
    - open the PSAP.

[CMPCOMSP] (**comp**leting **com**mitment with **sup**erior)

Completes the commitment processing for the branch with the superior.

- If ^Dsh
  - If Dg:
    - set Dc to TRUE;
    - set Dec to TRUE.
  - If ^Dg:
    - set Dc to FALSE;
    - set Dec to FALSE.

[CMPRB] (**comp**lete **r**oll**b**ack)

Performs the one-time actions to complete rollback of a transaction.

(Set Dec to Dc in polarized control mode.)

- If ^Dsh:
  - set Dec to Dc.

Recalculate variables for tree extension checks.

- If ^Dch or (Dsup and ^Nch)

  These branches are not in the transaction tree for the next transaction:

  - invoke [TREERESET].
- Subtract one unit from Ncntc.
- If ^Db:
  - open the PSAP.
- If Ncntc = 0

  Perform the one-time actions at the end of rollback:

  - issue a TP-ROLLBACK-COMPLETE ind;
  - set Nfa, Np, Nt, Nrpend, and Nfrb to FALSE.
  - If ^Nr and ^Ni and ^Nlf

    The node is no longer in a transaction tree:

    - set Ntch to FALSE.

  Activate NFSM if there is a branch for the next transaction, otherwise deactivate NFSM.

  - If Nr or Ni or Nlf:
    - generate "activate-nfsm".
  - If ^Nr and ^Ni and ^Nlf:
    - generate "deactivate-nfsm".

  Reset transaction completion request:

  - set Ncmtr to FALSE;
  - set Nopr to FALSE;
  - set Nror to FALSE;
  - set Neer to FALSE.

[COMREQ] (issue **com**mit **req**uest)

Invoked to issue the correct type of *commit request* or substitute to a single neighbour.

Issue the correct type of commit request on a dialogue with two-phase commit procedures.

- If Drdyi:
  - set D2pc to TRUE.

  Commit confirm awaited:

  - add one unit to Ncnt.
  - If ^Dsup
    - If ^Dhrsfu and ^Ldhrcomp

      Heuristic report awaited on the dialogue and needed for reporting to superior:

      - set Dnchra to TRUE.
    - If Dch

      If the superior dialogue is chaining and the *coordination level is commitment*, the aaid for the next transaction has been received with the *commit indication*, otherwise (*the coordination level is one-phase commitment*) the aaid has been received with the *ready-substitute indication* from the superior and a C-BEGIN ind is still awaited.

      - If Naaidn = NULL

        Commence a new transaction:

        - set Naaidn to a new unique value;
        - addNode (SnD, Naaidn, Nbridn).

```
          –  Set Dbridn to a unique value;
          –  set the atomic-action-identifier parameter to Naaidn;
          –  set the atomic-action-branch-identifier parameter to Dbridn;
          –  issue a C-COMMIT+C-BEGIN req;
          –  addBranch (SnD (Naaidn, Nbridn).sbbr, Dbridn, Aaet);
          –  set SnD (Naaidn, Nbridn).sbbr (Dbridn).rch to Arrh.
     –  If ^Dch and ^Dtb:
          –  issue a C-COMMIT req.
     –  If Dtb:
          –  issue a AF-ABORT (user,commitRI) req;
          –  set Dbpart to TRUE;
          –  set Dtb to FALSE.
–  If Dsup
     –  If ^Dhrsfu or Dcdfu
        Reporting applies on the dialogue with the superior (pending TP-U-ABORT req not issued).
        NOTE 3 – COMREQ is not used if Db is TRUE, i.e. if Dcdfu, then completion reporting is necessary.
          –  Issue C-COMMIT req.
     –  If Dhrsfu and ^Dcdfu
        No reporting on the dialogue.
          –  If ^Dtb:
               –  issue C-COMMIT req.
          –  If Dtb:
               –  issue a AF-ABORT (user,commitRI) req;
               –  set Dbpart to TRUE;
               –  set Dtb to FALSE.
```

Issue the correct type of C-NOCHANGE cnf on a dialogue with one-phase or read-only procedures.

```
–  If Dopi and ^Dsup and Du
   The chained functional unit cannot be selected, the dynamic commit functional unit is selected (one-phase uptree).
     –  If ^Dtb
          –  If Nresult = commit:
               –  issue C-NOCHANGE (commit) rsp.
          –  If Nresult = not-determined:
               –  issue C-NOCHANGE (not-determined) rsp.
     –  If Dtb
          –  If Nresult = commit:
               –  issue AF-ABORT (nochangeRC,commit) req.
          –  If Nresult = not-determined:
               –  issue AF-ABORT (nochangeRC,not-determined) req.
          –  Set Dbpart to TRUE.
          –  Set Dtb to FALSE.
```

- If Dopi and Dsup and Nch

  NOTE 4 – If a TP-U-ABORT req is pending, Danu is TRUE and the dialogue is no longer chaining, procedure NOTCHAIN already invoked in the calling subcell.

  C-BEGIN ind awaited for completion:

  - add one unit to Ncnt.
  - If Nresult = commit:
    - issue C-NOCHANGE (committed) rsp.
  - If Nresult = not-determined:
    - issue C-NOCHANGE (not-determined) rsp.
- If Dopi and Dsup and ^Nch

  NOTE 5 – If the chained transactions functional unit is selected on the dialogue, and a TP-U-ABORT req is pending, the abort is issued either with the result or with the report. The superior will not send C-BEGIN req.

  - If ^Danu
    - If Nresult = commit:
      - issue C-NOCHANGE (committed) rsp.
    - If Nresult = not-determined:
      - issue C-NOCHANGE (not-determined) rsp.
  - If Danu
    - If (Dhrsfu and ^Dcdfu) or (Nresult = not-determined)

      No reporting to superior awaited, abort issued with result (last PDU).

      - If Nresult = commit:
        - issue AF-ABORT (nochangeRC,commit) req.
      - If Nresult = not-determined:
        - issue AF-ABORT (nochangeRC,not-determined) req.
      - Set Dbpart to TRUE.
      - Set Danu to FALSE.
      - Set Dtb to FALSE.
    - If (^Dhrsfu or Dcdfu) and (Nresult = commit)

      Reporting to superior awaited, abort issued with report (last PDU):

      - issue C-NOCHANGE (committed) rsp.
- If Droi

  This is a subordinate dialogue and there are defer-actions pending. No rollback of the current transaction is possible. The result "not-determined" is issued to the subordinate.

  NOTE 6 – If the dialogue is chaining and there is no deferred-end pending, C-BEGIN req is issued in the action on completion of the transaction – late begin-.

  The procedure is not invoked if the dialogue is detached.

  - If ^Dtb:
    - issue C-NOCHANGE (not-determined) req.
  - If Dtb:
    - issue AF-ABORT (nochangeRC,not-determined) req;
    - set Dbpart to TRUE;
    - set Dtb to FALSE.

Procedure COMREQ is not used for a subordinate with early-exit, rollback on the branch is complete.

[COMRSP] (issue **com**mit **resp**onse)

Invoked to issue the correct type of *commit response* to the superior (if not already issued).

NOTE 7 – If the coordinator is a subordinate, the correct type of commit confirm is issued in procedure CMPCOMSB.

- If ^Ncc
  - If ^memsp (SldD, Naaid, Nbrid) or Dhrsfu
    - If Dtb and (Du or Danu)
      - If Dcdfu and (Ncrsev ^= EMPTY or Ncrud ^= EMPTY):
        - set *severity* parameter to Ncrsev;
        - set *completion-data* parameter to Ncrud;
        - issue an AF-ABORT-AND-REPORT (commitRC) req.
      - If ^Dcdfu or (Ncrsev = EMPTY and Ncrud = EMPTY):
        - issue an AF-ABORT (commitRC) req.
    - If ^Dtb or (Dtb and ^Du and ^Danu)
      - If Dcdfu and (Ncrsev ^= EMPTY or Ncrud ^= EMPTY):
        - set *severity* parameter to Ncrsev;
        - set *completion-data* parameter to Ncrud;
        - issue an AF-REPORT (commitRC) req.
    - If ^Dcdfu or (Ncrsev = EMPTY and Ncrud = EMPTY):
      - issue a C-COMMIT rsp.
  - If memsp (SldD, Naaid, Nbrid) and ^Dhrsfu
    - If Dcdfu and (Ncrsev ^= EMPTY or Ncrud ^= EMPTY):
      - set *severity* parameter to Ncrsev;
      - set *completion-data* parameter to Ncrud.
    - If Dtb and (Du or Danu):
      - set *heuristic-report* parameter to SldD (Naaid, Nbrid).type;
      - issue an AF-ABORT-AND-REPORT (commitRC) req.
    - If ^Dtb or (Dtb and ^Du and ^Danu):
      - set *heuristic-report* parameter to SldD (Naaid, Nbrid).type;
      - issue an AF-REPORT (commitRC) req.
  - If Dtb and Du:
    - set Dbpart to TRUE;
    - set Dtb to FALSE.

[COUNTCOM] (**count com**mitment confirm event)

Counts an event to complete commitment phase 2 processing for the node. Used to count a *commit confirm* or a TP-DONE request or a report on data (OCC-path) or a C-BEGIN indication (chaining static one-phase superior). When all events have been received, commences the tidying dialogues action necessary prior to commit completion on the node.

If the next transaction rolls back after commitment is complete, the counter is set to the number of subordinates from which a *rollback confirm* will be received plus the number of branches from which a TP-DONE request must be received.

If the next transaction does not rollback after commitment is complete, the counter Ncntge and Ncntrdy are set to the number of neighbours from which a C-READY indication or substitute will be received plus the number of branches from which a *transaction completion request* must be received.

The counter is adjusted during dialogue tidying for dialogues which are not to be included in the next transaction.

- Subtract one unit from Ncnt.
- If Ncnt = 0

  Delete current transaction, and commence tidying the dialogues:

  - delNode (SnD, Naaid, Nbrid);
  - set Nfa and Np to FALSE.

  Set commit completion counter to count down dialogue tidying:

  - set Ncntc to Nsubnb.
  - If Ni or Nlf:
    - add one unit to Ncntc.

  One unit for the NFSM:

  - add one unit to Ncntc.

  Initialize counters for the ready signals and ready-substitute indications:

  - set Nrdyi to 0;
  - set Nopi to 0;
  - set Ncntcdfu to 0.
  - If ^Nrpend

    Set the counters for phase one of commitment for the next transaction; one *ready signal* or *ready-substitute indication* from each neighbour; a *transaction completion request* for each branch.

    - If Nr or Ni:
      - set Ncntge to Nsubnb.
    - If Ni, add one unit to Ncntge.
    - If Nlf, set Ncntge to 1.

    One unit for the NFSM:

    - add one unit to Ncntge;
    - set Ncntrdy to Ncntge.
  - If Nrpend

    Set the counter to owe a TP-DONE req and a rollback confirm from each subordinate:

    - set Ncnt to Nsubnb.
    - If Nr or Ni:
      - add Nsubnb to Ncnt.
    - If Ni or Nlf:
      - add one unit to Ncnt.

    One unit for the NFSM:

    - add one unit to Ncnt.
- Generate "Complete-commit".

[COUNTCR] (**count c**ompletion **r**eport events)

After abort of the dialogue, no completion report is receivable. If a completion report was awaited, the counter is modified. A heuristic report is still receivable (if one is awaited) if there is recovery for the transaction branch (two-phase commit procedures).

- If (Nrpdhr or Nrpdcr) and ^Nr

  Reporting on data to the superior is necessary.

  - If Dcdfu:
    - subtract one unit from Ncntcr.

- If Ncntcr = 0:
  - set Ncrst to TRUE.
- If ((Nrpdhr and Ncnthr = 0) and (Nrpdcr and Ncntcr = 0))

  or ((Nrpdhr and Ncnthr = 0) and ^Nrpdcr)

  or (^Nrpdhr and (Nrpdcr and Ncntcr = 0))

  Subordinate reporting status is known at the node:

  either subordinate heuristic status and subordinate completion report status and both are necessary for reporting to superior,

  or only heuristic status is needed for reporting to superior and subordinate heuristic status is known,

  or only completion report status is needed for reporting to superior and subordinate completion report status is known.

  - If Dfdone and ^Dcrpa

    TPSUI reporting status is known, all reports collected:

    - set Nrpdhr to FALSE;
    - set Nrpdcr to FALSE;
    - generate "send-report".

[COUNTGE] (**count** **g**lobal **e**vent)

Counts the events needed to complete phase 1 of commitment; a *transaction completion request* to each branch.

When all events have been received, performs the one-time actions associated with completing phase 1 (in procedure DECISION).

If there was only one C-READY indication and there are no bound data at the node, the node either decides to send a C-NOCHANGE request on the dialogue with the C-READY indication (if possible) or to be the commit coordinator if sending is not possible. This decision depends on the state of the dialogue on which the one C-READY-RI was received and therefore the synchronizing event "one-ready" is used to activate this dialogue state machine.

If there is still one *ready signal* or *ready-substitute indication* missing, the synchronizing event "send-ready" is generated to find the appropriate state machine for propagation of the *ready signal* or *ready-substitute request* (if possible).

The counters Ncntrdy and Ncntge replace counter Ncnt as there are two different meanings for Ncntge + Ncntrdy = Ncnt = 1:

1) there is a *ready signal* or a *ready-substitute indication* outstanding; and
2) there is one dialogue state machine not having processed the *transaction completion request* and no *ready signal* or a *ready-substitute indication* is outstanding.

- Subtract one unit from Ncntge.
- If Nopr
  - If Dopfu:
    - add one unit to Ncntopfu.
  - If Ncntge = 0 and Ncntopfu = 0:
    - set the *diagnostic* parameter to "user-protocol-error";
    - generate "Rollback-by-TPPM";
    - exit this procedure.
  - If Ncntge = 0:
    - set Ncntopfu to 0.
- If (Ncntrdy = 0) and (Ncntge = 0)
  - If ^LDbd and (Nopr or Nror) and (Nrdyi = 1):
    - generate 'one-ready';
    - exit this procedure.

No forced collision possible, the node is the coordinator:

  - invoke [DECISION].

- If (Ncntrdy = 1) and (Ncntge = 0)

  If there is exactly one *ready signal* or *ready-substitute indication* missing, the state machine for this dialogue takes the necessary actions to enter the ready/one-phase/read-only state if this is possible due to static and dynamic constraints for sending a *ready signal* or a *ready-substitute request*. This state machine is triggered by the synchronizing event 'send-ready?'. All other state machines are in the state 20.02.

  - If Neer:
    - generate 'enter-early-exit-state'.
  - If ^Neer:
    - generate 'send-ready?'.

- If (Ncntrdy > 1) and (Ncntge = 0)

  If there is more than one *ready signal* or *ready-substitute indication* missing and the *transaction completion request* was performed by all state machines, sending of prepare is possible:

  - generate 'send-prepare'.

- If (Ncntrdy > 1) and (Ncntge > 0)

  If there is more than one *ready signal* or *ready-substitute indication* missing or the *transaction completion request* was not performed by all state machines, no action is taken:

  - continue.

[COUNTRB] (**count r**oll**b**ack confirm event)

Counts a *rollback confirm* event and performs the actions associated with completing rollback when all events have been received. When all events have arrived, the synchronizing event to report rollback to the superior is generated. If this is a root node, the one-time actions necessary to begin the next transaction are done including generating the synchronizing event.

- Subtract one unit from Ncnt.

Begin the next transaction for the root.

- If Ncnt = 0 and Nr:
  - delNode (SnD, Naaid, Nbrid);
  - set Naaid to NULL;
  - set Ncntge to Nsubnb.

  One unit for the NFSM:

  - add one unit ot Ncntge;
  - set Ncntrdy to Ncntge.

  Initialize counters for the *ready signals* and *ready-substitute indications*:

  - set Nrdyi to 0;
  - set Nopi to 0.
  - Set Ncntc to Nsubnb.

  Add one unit for NFSM:

  - add one unit to Ncntc;
  - set Ncntcdfu to 0;
  - generate "Complete-rollback".

Initiate reporting of rollback to the superior if not a root.

- If Ncnt = 0 and ^Nr:
  - generate "Report-rollback".

[COUNTRDY] (**count r**e**ady** events)

Counts the events needed to complete phase 1 of commitment; a *ready signal or a ready-substitute indication* from each neighbouring node (but one).

When all events have been received, performs the one-time actions associated with completing phase 1 (in procedure DECISION).

If there was only one C-READY indication and there are no bound data at the node, the node either decides to send a C-NOCHANGE request on the dialogue with the C-READY indication (if possible) or to be the commit coordinator if sending is not possible. This decision depends on the state of the dialogue on which the one C-READY-RI was received and therefore the synchronizing event "one-ready" is used to activate this dialogue state machine.

If there is still one *ready signal or ready-substitute indication* missing, the synchronizing event "send-ready?" is generated to find the appropriate state machine for propagation of the *ready signal or ready-substitute request* (if possible).

The counters Ncntrdy and Ncntge replace counter Ncnt as there are two different meanings for Ncntge + Ncntrdy = Ncnt = 1:

1) there is a *ready signal* or a *ready-substitute signal* outstanding; and

2) there is one dialogue state machine not having processed the *transaction completion request* and no *ready signal* or a *ready-substitute indication* is outstanding.

- Subtract one unit from Ncntrdy.
- If (Ncntrdy = 0) and (Ncntge = 0)
  - If ^LDbd and (Nopr or Nror) and (Nrdyi = 1):
    - generate 'one-ready';
    - exit this procedure.

  No forced collision possible, the node is the coordinator:

  - invoke [DECISION].
- If (Ncntrdy = 1) and (Ncntge = 0)

  If there is exactly one *ready signal* or *ready-substitute indication* missing, the state machine for this dialogue takes the necessary actions to enter the ready/one-phase/read-only state if this is possible due to static and dynamic constraints for sending a *ready signal* or a *ready-substitute request*. This state machine is triggered by the synchronizing event. All other state machines are in the state 20.2.

  - generate 'send-ready?'.
- If (Ncntrdy > 1) or (Ncntge > 0)

  If there is more than one *ready signal* or *ready-substitute indication* missing or the global event (*transaction completion request*) has not been received by all state machines, no action is taken:

  - continue.

[COUNTREP] (**count rep**orting status events)

Reporting (completion reporting and / or heuristic reporting) on the dialogue is complete (received or no longer receivable). This procedure is used by a state machine in state 21.1 and 21.2. After receiving the report (e.g. AF-HR indication, C-COMMIT confirm) or abort of a dialogue (e.g. A-P-ABORT indication) with one-phase commit procedures, the counter for the specific report type is modified if at least one report is awaited.

(Counting applies only for nodes on the OCC-path without root node.)

- If (Nrpdhr or Nrpdcr) and ^Nr

  Reporting on data to the superior is necessary.

  - If Dnchra:
    - subtract one unit from Ncnthr.
  - If Dcdfu and ^Db:
    - subtract one unit from Ncntcr.

- If Ncnthr = 0:
  - set Nhrst to TRUE.
- If Ncntcr = 0:
  - set Ncrst to TRUE.
- If ((Nrpdhr and Ncnthr = 0) and (Nrpdcr and Ncntcr = 0))

  or ((Nrpdhr and Ncnthr = 0) and ^Nrpdcr)

  or (^Nrpdhr and (Nrpdcr and Ncntcr = 0))

  subordinate reporting status is known at the node:

  either subordinate heuristic status and subordinate completion report status and both are necessary for reporting to superior,

  or only heuristic status is needed for reporting to superior and subordinate heuristic status is known,

  or only completion report status is needed for reporting to superior and subordinate completion report status is known.
- If Dfdone and ^Dcrpa

  TPSUI reporting status is known, all reports collected:
  - set Nrpdhr to FALSE;
  - set Nrpdcr to FALSE;
  - generate 'send-report'.

[COUNTREPDO] (**count rep**orting with TP-**DO**NE request)

This procedure is invoked by the superior dialogue state machine in state 21.5.1 after receiving TP-DONE request. If the reporting status is known (including the TP-DONE request), the report is issued after receiving "send-report" (generated this procedure and handled by this state machine in the actual action sequence). We use 'send-report' to find out the next main state for the superior dialogue state machine.

NOTE 8 – If the dialogue with the superior is aborted (p-abort) and no two-phase procedures are performed on the dialogue with the superior, then Nrpdhr is FALSE.

NOTE 9 – If the dialogue with the superior is aborted (p-abort), then Nrpdcr is FALSE.

- If Nrpdhr and Nhrst

  Reporting of heuristic status is necessary and possible.
  - If ^Nrpdcr

    Reporting of completion status is not necessary:
    - set Nrpdhr to FALSE;
    - generate 'send-report'.
  - If Nrpdcr and Ncrst

    Reporting of completion status is necessary and possible:
    - set Nrpdhr to FALSE;
    - set Nrpdcr to FALSE;
    - generate 'send-report'.
- If ^Nrpdhr

  Reporting of heuristic status is not necessary.
  - If Nrpdcr and Ncrst

    Reporting of completion status is necessary and possible:
    - set Nrpdcr to FALSE;
    - generate 'send-report'.

[CPSAP] (**c**lose the **PSAP**)

- Close the PSAP.

[CRDYRESET] (**C**-**R**EA**DY** indication received, **reset** the readyk field for the branch)

The readyk field of the branch identification of the variable Snd is set to none if a collision of a *ready signal* and a *ready-substitute indication* occurs on a dialogue (forced and unforced).

- If Dsup:
  - set SnD (Naaid, Nbrid).spbr.readyk to none.
- If ^Dsup:
  - set SnD (Naaid, Nbrid).sbbr(Dbrid).readyk to none.

[CRDYSET] (**C**-**R**EA**DY** indication received, **set** the readyk field for the branch)

The readyk field of the branch identification of the variable Snd is set to ready-received.

NOTE 10 – The appropriate action for ready-sent is done in procedures SEND2PC.

- If Dsup:
  - set SnD (Naaid, Nbrid).spbr.readyk to ready-received.
- If ^Dsup:
  - set SnD (Naaid, Nbrid).sbbr(Dbrid).readyk to ready-received.

[CRERSRT] [issue **C**-**RE**COVER (**ret**ry-later) **r**espon**s**e]

Invoked to issue a C-RECOVER (retry-later) response for the dialogue with a subordinate or with a superior.

- If Dsup:
  - set the *atomic-action-identifier* parameter to Naaid;
  - set the *atomic-action-branch-identifier* parameter to Nbrid;
  - issue C-RECOVER (retry-later) rsp.
- If ^Dsup:
  - set the *atomic-action-identifier* parameter to Naaid;
  - set the *atomic-action-branch-identifier* parameter to Dbrid;
  - issue C-RECOVER (retry-later) rsp.

[CRNALL] (**c**ompletion **r**eport parameters **n**ot **all**owed with next TP-DONE request)

- Set Ntbicr to FALSE.
- If Dcrpa:
  - generate 'cr-not-allowed'.

[DECDENB] (**Dec**rement **Denb**)

Invoked to decrement Denb and also Denbb, whenever applicable.

- Subtract one unit from Denb.
- If Denbb > 0:
  - subtract one unit from Denbb.

[DECISION] (the node is the commit coordinator)

This procedure is invoked if the node is the commit coordinator (Ncntrdy = 0) and (Ncntge = 0).

a) If the last *ready signal* or *ready-substitute indication* has not been sent. The state machine was in state 20.1 waiting for the last *ready signal* or *ready-substitute indication*.

b) If only one *ready signal* has been received and the node is not able to send a *ready-substitute indication* on this dialogue. The state machine is in state 20.1. The procedure is called from SNDORDCD.

c) If a collision of *ready signals* and/or *ready-substitute indications* occurs and the node is the commit coordinator (state 20.3).

- Set Ncnt to '0'.
- If (Nrdyi = 0) and ^Ldbd and (Nror or Nopr)

  No bound data and no C-READY ind and TPSUI is one-phase or read-only.

  - If (Nopi = 0) and Nopr

    The node is a read-only commit coordinator [there was no *ready signal* and no C/AF-NOCHANGE (result-required) ind, there are no modified bound data at the TPPM resp. at the TPSUI (this must be the root of the transaction tree)]:

    - set Nresult to 'commit';
    - issue TP-COMMIT ind;
    - generate 'continue-commit';
    - invoke [OWEDONECO];
    - exit this procedure.
  - If (Nopi > 0) and (Nopr or Nror) and ^Ldfail

    The node is a one-phase commit coordinator (there was no *ready signal*, there are no modified bound data at the TPPM resp. at the TPSUI and no rollback is to be initiated):

    - set Nresult to 'commit';
    - issue TP-COMMIT ind;
    - generate 'continue-commit';
    - invoke [OWEDONECO];
    - exit this procedure.

    NOTE 11 – If the Ldfail is true, the node decides to rollback the transaction although there are no real bound data, next action is '– if Ldfail' at the end of this procedure.

The node is a 2PC commit coordinator, even if only the TPPM or only the TPSUI has bound data.

- If ((Nrdyi > 0) or LDbd or Ncmtr) and ^Ldfail

  The node is able to write the commit-record:

  - set Nresult to 'commit';
  - set Snd (Naaid, Nbrid).type to "log-commit".

  If a heuristic decision was taken by the TPPM (the TPPM was in the READY-state if there was a collision of *ready signals*), there is a log-heuristic record. Create the log-damage record.

  - If memsp(SlhD,Naaid,Nbrid)
    - If ((SlhD,(Naaid,Nbrid).type ^= "heuristic-final"):
      - addNode(SldD,(Naaid,Nbrid).type to "heuristic-mix".
  - Begin setting the bound data to the final state unless a heuristic decision has been taken; the *TPPM bound data* shall eventually be set to the final state; when this occurs is a local matter.

    NOTE 12 – Although the node is the coordinator, a heuristic decision is possible if the node was in ready state (collision of ready signals).

  Issue decision to the TPSUI and generate a synchronizing event for the state machines:

  - issue TP-COMMIT ind;
  - generate 'continue-commit';
  - invoke [OWEDONECO].

If the node is not able to write the commit-record or there is a local decision to initiate rollback, a rollback is initiated.

- If Ldfail:
  - generate "rollback-by-tppm".

[DEFREQ] (send AF-**DEF**ER **req**uest)

Issues an AF-DEFER request that has been held pending receipt of a TP-PREPARE request or a *transaction completion request* from the TPSUI.

- If De:
  - issue an AF-DEFER (end-dialogue) req.
- If ^De and Dg:
  - issue an AF-DEFER (grant-control) req.

[DELBR] (**del**ete **br**anch)

Removes a transaction branch and adjusts the node variables accordingly.

- Set Dl to FALSE.
- If Dsup
  - If Ni:
    - set Ni to FALSE;
    - set Nr to TRUE.
  - If Nlf:
    - set Nlf to FALSE.
- If ^Dsup:
  - subtract one unit from Nsubnb;
  - set Dch to FALSE.
  - If Nsubnb = 0 and Ni:
    - set Ni to FALSE;
    - set Nlf to TRUE.
  - If Nsubnb = 0 and Nr:
    - set Nr to FALSE.

Reduce the counter for the *transaction completion request* and *ready signal* and *ready-substitute indication* if the next transaction is not to be rolled back. Otherwise reduce counter Ncnt, which is used during rollback.

- If ^Nrpend:
  - subtract one unit from Ncntge;
  - subtract one unit from Ncntrdy.
- If Nrpend

  One unit for rollback confirm:

  - subtract one unit from Ncnt.

  One unit for TP-DONE req:

  - subtract one unit from Ncnt.

[DELBRANCH] (**del**ete **branch** in system variable)

Removes a transaction branch from sbbr for a dialogue which is terminated during the ACTIVE state without causing a rollback.

- delBranch (SnD (Naaid, Nbrid).sbbr, Dbrid).
- If Dcdfu:
  - subtract one unit from Ncntcdfu.

NOTE 13 – The dialogue state machine is now in state 25 (zombie) and the counters Ncntge and Ncntrdy are modified when a *transaction completion request* occurs. The dialogue (state machine) is still corrdinated (although the dialogue is aborted), i.e. Dl is TRUE (for processing of global events and synchronizing events). A zombie dialogue is treated as a read-only subordinate.

[DELBRO] (**del**ete **b**ranch after **r**ead-**o**nly or early-exit)

After receiving C-READ-ONLY indication or AF-EARLY-EXIT indication on a branch with chaining transaction branches and no pending defer action or on a branch with unchained transactions, the subordinate node is no longer part of the transaction tree. The branch is to be deleted (procedure [DELBRANCH] is used).

- Set Dl to FALSE;
- subtract one unit from Nsubnb;
- set Dch to FALSE.
- If Nsubnb = 0 and Ni:
  - set Ni to FALSE;
  - set Nlf to TRUE.
- If Nsubnb = 0 and Nr:
  - set Nr to FALSE.

Reduce counter for *transaction completion request* if not yet received.

- If Ncntge > 0:
  - subtract one unit from Ncntge.

The counter Ncntrdy will be reduced in COUNTRDY (called in this subcell).

[DELIMIT] (**delimit** dialogue)

Handles the AF-BEGIN-DIALOGUE (accepted, dataRI) response which occurs before any requests are issued by the subordinate.

- If Dsup and ^Da:
  - issue an AF-BEGIN-DIALOGUE (accepted, dataRI) rsp;
  - set Ncr to FALSE;
  - set Da and Nrn to TRUE.
  - If Dl:
    - issue a C-BEGIN rsp.
  - While (Depnb > 0):
    - issue AF-U-ERROR rsp;
    - subtract one unit from Depnb.

[DELNBRANCH] (**del**ete **n**ext **branch** in system variable)

Removes a transaction branch from sbbr for a chained dialogue which is terminated during commitment without the atomic-action identifier of the next transaction having been used.

- delBranch (SnD (Naaidn, Nbridn).sbbr, Dbridn).

[DIALOGUE] (operate on **dialogue**)

Executes all subsequent actions on the dialogue.

[DISCARDS] (**discard s**eparator PDUs)

Invoked when initiating rollback.

- Discard all PDUs in the separator.

[EARLYC] (**early** commit **c**onfirmation)

Invoked if a commit response is sendable on the dialogue prior completion of the current transaction.

A commit response is issued on the dialogue or on the channel if a intermediate log commit record is written at the node (Nclw).

- If Nclw
  - If Dsup:
    - invoke [COMRSP].

- If ^Dsup
  - If Dtb and ^Db:
    - issue an AF-ABORT (commitRC) req;
    - set Dbpart to TRUE;
    - set Dtb to FALSE;
    - close the PSAP.
  - If ^Dtb and ^Db:
    - issue a C-COMMIT rsp;
    - close the PSAP.
  - If Db and Dchat:
    - invoke [RECVRDONE];
    - issue a CAF-DETACH (free) req;
    - set Dchat to FALSE.
- Set Ncc to TRUE.

[GENPREP] (**gen**erate AF-**PREP**ARE request with termination request)

Invoked if a *transaction completion request* has been received and *ready signal* or *ready-substitute request* is not sendable and no AF-PREPARE request has been issued and no *ready signal* or *ready-substitute indication* has been received. If sending of prepare is not necessary, the local decsion Ldprep is used to send (or not to send) a prepare signal.

- If Drrec
  - If (^Dsup and ^Dimpl) or (^Dsh and ^Ddp)
    - If Dsh:
      - issue an AF-PREPARE req.
    - If ^Dsh:
      - issue an AF-PREPARE (data-permitted = FALSE) req.
    - Set Dps to TRUE.
- If ^Dps and Ldprep
  - If Dsh:
    - issue an AF-PREPARE req.
  - If ^Dsh:
    - issue an AF-PREPARE (data-permitted = FALSE) req.
  - Set Dps to TRUE.

[INITDIASB] (**init**ialize **dia**logue with **sub**ordinate)

Invoked when an AF-BEGIN-DIALOGUE request is issued.

- If Handshake functional unit is selected:
  - set Dh to TRUE.
- If Unchained Transactions functional unit is selected:
  - set Du to TRUE.
- If Shared Control functional unit is selected:
  - set Dsh to TRUE.
  - If Du or Chained Transactions functional unit is selected:
    - set Dc to TRUE.
- If ^Dsh:
  - set Dec to TRUE.
- If Cancel functional unit is selected:
  - set Dcancfu to TRUE.

- If Completion Diagnostics functional unit is selected:
  - set Dcdfu to TRUE.
- If Dynamic Commit functional unit is selected:
  - set Ddyn to TRUE.
- If Early Exit functional unit is selected:
  - set Deefu to TRUE.
- If Heuristic Containment Required functional unit is selected:
  - set Dhrsfu to TRUE.
- If Implicit Prepare functional unit is selected:
  - set Dimpl to TRUE.
- If Commit functional unit is selected:
  - set Do to TRUE.
- If One-phase Commit functional unit is selected:
  - set Dopfu to TRUE.
- If Read-only functional unit is selected:
  - set Drofu to TRUE.
- If Ddyn and (Subordinate-may-send-ready = TRUE):
  - set Drrec to TRUE.
- If Do and ^Ddyn:
  - set Drrec to TRUE.
- If Ddyn and (Superior-may-send-ready = TRUE):
  - set Drsen to TRUE.
- If ^Do and Dopfu

  Static one-phase subordinate:
  - set Drsen to TRUE.
- If (*confirmation* = "always"):
  - set Dcr to TRUE.

[INITDIASP] (**init**ialize **dia**logue with **sup**erior)

Invoked when an AF-BEGIN-DIALOGUE indication is received.

- If Handshake functional unit is selected:
  - set Dh to TRUE.
- If Unchained Transactions functional unit is selected:
  - set Du to TRUE.
- If Shared Control functional unit is selected:
  - set Dsh to TRUE.
- If Du or Chained Transactions functional unit is selected:
  - set Dc to TRUE.
- If ^Dsh:
  - set Dec to FALSE.
- If Cancel functional unit is selected:
  - set Dcancfu to TRUE.
- If Completion Diagnostics functional unit is selected:
  - set Dcdfu to TRUE.
- If Dynamic Commit functional unit is selected:
  - set Ddyn to TRUE.

- If Early Exit functional unit is selected:
  - set Deefu to TRUE.
- If Heuristic Containment Required functional unit is selected:
  - set Dhrsfu to TRUE.
- If Implicit Prepare functional unit is selected:
  - set Dimpl to TRUE;
  - set Nimpl to TRUE;
  - set Np to TRUE.
- If Commit functional unit is selected:
  - set Do to TRUE.
- If One-phase Commit functional unit is selected:
  - set Dopfu to TRUE.
- If Read-only functional unit is selected:
  - set Drofu to TRUE.
- If (Ddyn and Subordinate-may-send-ready = TRUE):
  - set Drsen to TRUE.
- If Do and ^Ddyn:
  - set Drsen to TRUE.
- If Ddyn and Superior-may-send-ready = TRUE:
  - set Drrec to TRUE.
- If ^Do and Dopfu

  ready-substitute indication is receivable on a static one-phase superior branch:
  - set Drrec to TRUE.
- Set Dsup to TRUE.
- If (*confirmation* = "always"):
  - set Ncr to TRUE.

[INITMACF] (**init**ialize **MACF**)

Invoked upon receipt of an AF-BEGIN-DIALOGUE indication at a leaf node, and on the first TP-BEGIN-DIALOGUE request from a root node.

- Set all node variables to their initial values, as specified in A.4.2.4.

[INITRB] (**init**iate **r**oll**b**ack)

Initiates rollback at this node. Sets the counter to account for a *rollback confirm/indication* from each subordinate.

- Set the bound data to the initial state;
- set Nfrb to TRUE;
- set Ncnt to Nsubnb;
- set Nt to TRUE;
- set SnD (Naaid, Nbrid).type to NULL;
- generate "Rollback-all".
- If Ntbicr:
  - generate 'cr-allowed'.

[**INITREPSP**] (**init**ialize **rep**orting on the OCC-path, **sup**erior dialogue)

Reporting on data applies on the dialogue with the superior.

This Procedure is used during processing of 'continue-commit' on the dialogue to the superior.

- If ^Dhrsfu and (^Db or D2pc)

  Heuristic reporting necessary:

  - set Nrpdhr to TRUE.
- If Dcdfu and ^Db

  Completion reporting necessary:

  - set Nrpdcr to TRUE.
- Set Ncntcr to 0.
- Set Ncnthr to 0.
- Set Nhrst to TRUE.
- Set Ncrst to TRUE.
- Generate 'report-status'.

[LOGDAM] (**log dam**age)

Updates the log-damage record according to the value of the *heuristic-report* parameter of either a TP-DONE request or an AF-[ABORT-AND-]REPORT indication.

If this is a compensatable heuristic report, no changes are made.

- If Dnchra
  - If the value of the heuristic-report parameter is "heuristic-hazard"
    - If ^memsp (SldD, Naaid, Nbrid):
      - addNode (SldD, Naaid, Nbrid);
      - set SldD (Naaid, Nbrid).type to "heuristic-hazard".
  - If the value of the heuristic-report parameter is "heuristic-mix"
    - If ^memsp (SldD, Naaid, Nbrid):
      - addNode (SldD, Naaid, Nbrid);
      - set SldD (Naaid, Nbrid).type to "heuristic-mix".
    - If memsp (SldD, Naaid, Nbrid)
      - If (SldD (Naaid, Nbrid).type = "heuristic-hazard"):
        - set SldD (Naaid, Nbrid).type to "heuristic-mix".

[LOGDAMH] (**log dam**age **h**azard)

Creates a the log-damage record with the value of "heuristic-hazard".

If this is a compensatable heuristic report, no changes are made. [Example: A-P-ABORT indication on a subordinate dialogue after sending AF-PREPARE request causes a heuristic-hazard, but this might be a read-only subordinate and this is known at the local system (the subordinate was ordered to go read-only).]

- If Dnchra or (^Dnchra and ^Ldhrcomp)

  The procedure used in active state too, Dnchra is always FALSE and Ldhrcomp must be used.

  - If ^memsp (SldD, Naaid, Nbrid):
    - addNode (SldD, Naaid, Nbrid);
    - set SldD (Naaid, Nbrid).type to "heuristic-hazard".

[LOGDAMRB] (**log dam**age **r**oll**b**ack)

Updates the log-damage as necessary in the event of a rollback.

- If memsp (SlhD, Naaid, Nbrid)
  - If SlhD (Naaid, Nbrid).type ^= "heuristic-initial":
    - addNode (SldD, Naaid, Nbrid);
    - set SldD (Naaid, Nbrid).type to "heuristic-mix".

[LOGHD] (**log h**euristic **d**ecision)

Updates the log-heuristic record according to the value of the *heuristic-report* parameter from the heuristic decision.

- If ^memsp (SlhD, Naaid, Nbrid):
  - addNode (SlhD, Naaid, Nbrid);
  - set SlhD (Naaid, Nbrid).type to the *heuristic-report* parameter.
- If memsp (SlhD, Naaid, Nbrid)
  - If (SlhD (Naaid, Nbrid).type ^= *heuristic-report*):
    - set SlhD (Naaid, Nbrid).type to "heuristic-mix".

[LOGREMOVE] (**log**-heuristic,log-damage **remove**)

Removes the log-damage and log-heuristic records.

- delNode (SldD, Naaid, Nbrid).
- If memsp (SlhD, Naaid, Nbrid):
  - delNode (SlhD, Naaid, Nbrid).

[NEWCHANNEL] (operate on **new channel**)

Executes all subsequent actions in this subcell on the channel that is attached to the TPPM with the CAF-RECOVER indication

[NOTCHAIN] (**not chain**ing)

Invoked when the dialogue is no longer *chaining transaction branches*.

- If Dsup:
  - set Nch to FALSE.
- If ^Dsup:
  - set Dch to FALSE.

[NXTBR] (commence **next br**anch in chain)

Commences the next branch for a chaining subordinate dialogue.

- If Naaid = NULL

  Commence a new Atomic-action:
  - set Naaid to a new unique value;
  - set Nbrid to a new unique value;
  - addNode (SnD, Naaid, Nbrid).
- Set Dbrid to a new unique value.
- AddBranch(SnD (Naaid, Nbrid).sbbr, Dbrid, Aaet).
- Set SnD(Naaid, Nbrid).sbbr (Dbrid).rch to Arrh.

[NXTTRAN] (**next tran**saction)

Does the one-time actions for a leaf or intermediate node when rollback is complete and the next transaction after a rollback is to begin.

- If Ni:
  - set Ncntge to Nsubnb;
  - set Ncntc to Nsubnb.
- If Nlf:
  - set Ncntge to 0;
  - set Ncntc to 0.

One unit for NFSM:

- add one unit to Ncntc;
- add one unit to Ncntge.

One unit for the superior:

- add one unit to Ncntc;
- add one unit to Ncntge;
- set Ncntrdy to Ncntge.

Initialize counters for the ready signals and *ready-substitute indications*:

- set Nrdyi to 0;
- set Nopi to 0;
- set Ncntcdfu to 0.
- If ^Nch:
  - delNode (SnD, Naaid, Nbrid);
  - set Naaid to NULL;
  - set Nbrid to NULL.

If superior dialogue is chaining, the existing node will have been deleted and a new Atomic-action brought into use by [CBEAFTRB]:

- Set Ncnt to 1.

  NOTE 14 – The condition "Completion of the transaction is necessary although Ncnt > 0" is only fulfilled after having received C-ROLLBACK cnf as a response to AF-EARLY-EXIT req on a chaining superior dialogue. The rollback procedures are used at the node and C-BEGIN ind is awaited. The counter is set to "1" and the counter will be decremented by the NFSM when processing "complete-rollback".

- Generate "Complete-rollback".

[OLDCHANNEL] (operate on **old channel**)

Executes all subsequent actions in this subcell on the channel that was already attached to the TPPM before a CAF-RECOVER indicationwas received on a different channel.

[OPSAP] (**o**pen **PSAP**)

- Open the PSAP.

[OWEDONE] (**owe** a TP-**DONE** request)

Makes a *TP-DONE request owed* and allows failure-related actions. If a *TP-DONE request is not owed,* adjust the count of events expected.

- Set Nfa to TRUE.
- If ^Dd:
  - generate "Set-done-true".
  - If ^Drbrep
    - If Nr or Ni, add Nsubnb to Ncnt,
    - If Ni or Nlf, add one unit to Ncnt.

  One unit for NFSM:

  - add one unit to Ncnt.

[OWEDONECO] (**owe** a TP-**DONE** request after **co**mmit indication and adjust the count of events expected)

Makes a *TP-DONE request owed*.

- Generate "Set-done-true".
- If Nr or Ni:
  - add Nsubnb to Ncnt.
- If Ni or Nlf:
  - add one unit to Ncnt.

One unit for NFSM:

- add one unit to Ncnt.

[PREPREQ] (issue AF-**PREP**ARE **req**uest)

- If Dsh:
  - issue an AF-PREPARE req.
- If ^Dsh:
  - issue an AF-PREPARE (data-permitted = FALSE) req.

[RBNEXTSB] (**r**oll**b**ack **next** transaction **sub**ordinate)

Invoked when it is determined that the next transaction will rollback and the dialogue is chaining.

- If ^Nrpend:
  - set Nrpend to TRUE;
  - generate "Rollback-next-trans".

[RBREQ] (issue **r**oll**b**ack **req**uest)

Invoked to issue a *rollback request* of the correct form.

- If ^Dsup
  - If ^Dtb:
    - issue C-ROLLBACK req.
  - If Dtb:
    - issue an AF-ABORT (user, rollbackRI) req.
- If Dsup
  - If ^memsp (SldD, Naaid, Nbrid) or Dhrsfu or (^Nimpl and ^Np)
    - If (Ncrsev ^= EMPTY or Ncrud ^= EMPTY) and Dcdfu:
      - set *severity* parameter to Ncrsev;
      - set *completion-data* parameter to Ncrud;
      - issue an AF-REPORT (rollbackRI) req.
    - If (Ncrsev = EMPTY and Ncrud = EMPTY) or ^Dcdfu
      - If ^Dtb:
        - issue C-ROLLBACK req.
      - If Dtb:
        - issue an AF-ABORT (user, rollbackRI) req.

  - If memsp (SldD, Naaid, Nbrid) and ^Dtb and ^Dhrsfu and (Nimpl or Np)
    - If (Ncrsev ^= EMPTY or Ncrud ^= EMPTY) and Dcdfu:
      - set *severity* parameter to Ncrsev;
      - set *completion-data* parameter to Ncrud.
    - Set *heuristic-report* parameter to SldD (Naaid, Nbrid).type.
    - Issue an AF-REPORT (rollbackRI) req.
  - If memsp (SldD, Naaid, Nbrid) and Dtb and ^Dhrsfu and (Nimpl or Np)
    - If (Ncrsev ^= EMPTY or Ncrud ^= EMPTY) and Dcdfu:
      - set *severity* parameter to Ncrsev;
      - set *completion-data* parameter to Ncrud.
    - Set *heuristic-report* parameter to SldD (Naaid, Nbrid).type.
    - Issue an AF-ABORT-AND-REPORT (rollbackRI) req.
- If Dtb:
  - set Dbpart to TRUE;
  - set Dtb to FALSE.

[RBRSPNOAB] (issue **r**oll**b**ack **resp**onse **no** **ab**ort)

Invoked to issue a *rollback response* if no abort is included.

- If ^Dsup:
  - issue C-ROLLBACK rsp.
- If Dsup
  - If ^memsp (SldD, Naaid, Nbrid) or Dhrsfu or (^Nimpl and ^Np)
    - If (Ncrsev ^= EMPTY or Ncrud ^= EMPTY) and Dcdfu:
      - set *severity* parameter to Ncrsev;
      - set *completion-data* parameter to Ncrud;
      - issue an AF-REPORT (rollbackRC) req.
    - If (Ncrsev = EMPTY and Ncrud = EMPTY) or ^Dcdfu:
      - Issue C-ROLLBACK rsp.
  - If memsp (SldD, Naaid, Nbrid) and ^Dhrsfu and (Nimpl or Np):
    - set *heuristic-report* parameter to SldD (Naaid, Nbrid).type.
    - If (Ncrsev ^= EMPTY or Ncrud ^= EMPTY) and ^Dcdfu:
      - set *severity* parameter to Ncrsev;
      - set *completion-data* parameter to Ncrud.
    - Issue an AF-REPORT (rollbackRC) req.

[RBRSPAB] (issue **r**oll**b**ack **resp**onse with **ab**ort)

Invoked to issue a *rollback response* together with a user abort. Either Dtb or Dbpart is TRUE when this procedure is invoked.

- If ^Dsup:
  - issue an AF-ABORT (user, rollbackRC) req.
- If Dsup
  - If ^memsp (SldD, Naaid, Nbrid) or Dhrsfu or (^Nimpl and ^Np)
    - If (Ncrsev ^= EMPTY or Ncrud ^= EMPTY) and Dcdfu:
      - set *severity* parameter to Ncrsev;
      - set *completion-data* parameter to Ncrud;
      - issue an AF-REPORT (rollbackRC) req.

- If (Ncrsev = EMPTY and Ncrud = EMPTY) or ^Dcdfu:
  - issue an AF-ABORT (user, rollbackRC) req.
- If memsp (SldD, Naaid, Nbrid) and ^Dhrsfu and (Nimpl or Np):
  - set *heuristic-report* parameter to SldD (Naaid, Nbrid).type.
  - If (Ncrsev ^= EMPTY or Ncrud ^= EMPTY) and Dcdfu:
    - set *severity* parameter to Ncrsev;
    - set *completion-data* parameter to Ncrud.
  - Issue an AF-ABORT-AND-REPORT (rollbackRC) req.

[RECCOM] (**rec**eive **com**mit indication)

Receives the *commit indication*. Sets the counter to account for receiving a *commit confirm* from each (2PC) neighbour.

- Set Ncnt to 0;
- set Nresult to 'commit';
- set D2pc to TRUE;
- issue a TP-COMMIT ind;
- begin setting the *TPPM bound data* to the final state, unless a heuristic decision has been taken; the *TPPM bound data* shall eventually be set to the final state; when this occurs is a local matter.
- If Nch and ^Danu and Dsup:
  - set Naaidn to the *atomic-action-identifier* parameter;
  - set Nbridn to the *atomic-action-branch-identifier* parameter;
  - addNode(SnD, Naaidn, Nbridn);
  - set SnD(Naaidn, Nbridn).spbr.rch to Arrh.
- If memsp (SlhD, Naaid, Nbrid)
  - If (SlhD (Naaid, Nbrid).type ^= "heuristic-final"):
    - addNode(SldD, Naaid, Nbrid);
    - set SldD (Naaid, Nbrid).type to "heuristic-mix".
- If ^Dsup

  The node is on the OCC-path.

  - If ^Dhrsfu or (Dcdfu and ^Db)

    The report on data from the subordinate is needed for completion of commitment (instead of a commit confirm):

    - add one unit to Ncnt.
  - If ^Dhrsfu and Ldhrcomp:
    - set Dnchra to TRUE.

      NOTE 15 – Reporting status is not known at the node, first TP-DONE req is awaited.

    - Generate "Continue-commit".

[RECCOM-OP] (**rec**eive **com**mit order if the node is **o**ne-**p**hase or read-only)

Receives the outcome of the transaction in the one-phase or the read-only-state. The node enters the decided-one-phase or decided-read-only-state. Nresult is either commit, not-determined or no-change (set in calling subcell). The result value 'no-change' is only used in this procedure and set to 'commit' for the following reason: a report is only awaited if the result is 'commit' but after having ajusted the counter for reporting (see below), the result 'no-change' is handled as the result 'commit'.

- Set Ncnt to 0.

Calculate the resulting message for the TPSUI.

- If Nresult = 'commit' or Nresult = 'no-change':
    - issue a TP-COMMIT ind;
    - generate 'continue-commit'.
- If Nresult = 'not-determined':
    - issue TP-UNKNOWN ind;
    - generate 'continue-unknown'.
- If ^Dsup

  The node is on the OCC-path.
    - If (^Dhrsfu or (Dcdfu and ^Db)) and (Nresult = commit)

      The report on data from the subordinate is needed for completion of commitment:
        - add one unit to Ncnt.
    - If ^Dhrsfu and Ldhrcomp:
        - set Dnchra to TRUE.

          NOTE 16 – Reporting status is not known at the node, first TP-DONE req is awaited.
- If Nresult = 'no-change':
    - set Nresult to 'commit'.

[RECVRCOMI] [issue C-**REC**O**V**E**R** (**com**m**i**t) request]

Invoked to issue the correct type of C-RECOVER (commit) request.

Heuristic reporting is not yet possible on the channel with the superior (heuristic status not known). The heursitic report parameter is absent. If the heuristic status is known, AF-REPORT (commit) request is issued in procedure SENDREP?.

This procedure is only used in response to a local channel request (CAF-PLEASE request / CAF-GIVE indication). The recovery context handle is specified. In response to a recovery indication from a neighbour, the recovery context handle is not specified and procedure RECVRCOMR is used.

- Set the *atomic-action-identifier* parameter to Naaid.
- If ^Dsup:
    - set the atomic-action-branch-identifier parameter to Dbrid.
    - If SnD(Naaid, Nbrid).sbbr (Dbrid).rch = NULL:
        - issue C-RECOVER (commit) req.
    - If SnD(Naaid, Nbrid).sbbr (Dbrid).rch ^= NULL:
        - set the *recovery-context-handle* parameter to SnD (Naaid, Nbrid).sbbr (Dbrid).rch;
        - issue an AF-RECOVER (commit) req.
- If Dsup:
    - set the atomic-action-branch-identifier parameter to Nbrid.
    - If SnD (Naaid, Nbrid).spbr.rch = NULL:
        - issue C-RECOVER (commit) req.
    - If SnD (Naaid, Nbrid).spbr.rch ^= NULL:
        - set the *recovery-context-handle* parameter to SnD (Naaid, Nbrid).spbr.rch;
        - issue an AF-RECOVER (commit) req.
- If Atwr:
    - issue an AF-TOKEN-GIVE (two-way-recovery) req.

[RECVRCOMR] [issue C-**REC**OVE**R** (**com**mit) **r**equest after CAF-RECOVER (ready) indication]

Invoked to issue C-RECOVER (commit) request when a CAF-RECOVER (ready) indication was received.

- Set the *atomic-action-identifier* parameter to Naaid.
- If ^Dsup:
    - set the *atomic-action-branch-identifier* parameter to Dbrid;
    - issue a C-RECOVER (commit) req.
- If Dsup:
    - set the atomic-action-branch-identifier parameter to Nbrid;
    - issue C-RECOVER (commit) req.

[RECVRDONE] [issue C-**REC**OVE**R** (**done**) response]

Invoked to issue the correct type of C-RECOVER (done) response.

- Set the *atomic-action-identifier* parameter to Naaid.
- If Dsup:
    - set the atomic-action-branch-identifier parameter to Nbrid.
    - If Dhrsfu:
        - issue C-RECOVER (done) rsp.
    - If ^Dhrsfu
        - If memsp (SldD, Naaid, Nbrid):
            - set the heuristic-report parameter to SldD (Naaid, Nbrid).type.
        - If ^memsp (SldD, Naaid, Nbrid):
            - set the heuristic-report parameter to 'none'.
        - Issue an AF-REPORT (recoverDoneRC) req.
- If ^Dsup:
    - set the atomic-action-branch-identifier parameter to Dbrid;
    - issue C-RECOVER (done) rsp.

[RECVRRDY] [issue C-**REC**OVE**R** (**r**e**ady**) request]

Invoked to issue the correct type of C-RECOVER (ready) request.

- Set the *atomic-action-identifier* parameter to Naaid.
- If Dsup:
    - set the atomic-action-branch-identifier parameter to Nbrid.
    - If SnD (Naaid, Nbrid).spbr.rch = NULL:
        - issue C-RECOVER (ready) req.
    - If SnD (Naaid, Nbrid).spbr.rch ^= NULL:
        - set the *recovery-context-handle* parameter to SnD (Naaid, Nbrid).spbr.rch;
        - issue an AF-RECOVER (ready) req.
- If ^Dsup:
    - set the atomic-action-branch-identifier parameter to Dbrid.
    - If SnD (Naaid, Nbrid).sbbr(Dbrid).rch = NULL:
        - issue C-RECOVER (ready) req.
    - If SnD (Naaid, Nbrid).sbbr(Dbrid).rch ^= NULL:
        - set the *recovery-context-handle* parameter to SnD (Naaid, Nbrid).sbbr(Dbird).rch;
        - issue an AF-RECOVER (ready) req.
- If Atwr:
    - issue an AF-TOKEN-GIVE (two-way-recovery) req.

[REJTRAN] (**rej**ect **tran**saction)

Removes a transaction branch on the dialogue with the superior, the transaction node, and adjusts the node variables accordingly.

- Set Dl to FALSE;
- set Nch to FALSE;
- set Nlf to FALSE.

Reduce the counter for the C-READY ind:

- subtract one unit from Ncntge;
- subtract one unit from Ncntrdy;
- delNode(SnD, Naaid, Nbrid).

The node finite state machine is no longer active, there is no transaction at the node:

- generate "deactivate-nfsm".

[RESETAAIDN] (**reset** the **a**tomic **a**ction **id**entifier for the **n**ext transaction)

The atomic action identifier and the branch identifier received with the C-NOCHANGE (result-requested) indication (on the chaining superior dialogue) are no longer valid and are to be deleted.

- Set Naaidn to NULL;
- set Nbridn to NULL.

[RESETD] (**reset d**ialogue variables)

Resets dialogue variables for the next transaction.

- Set Dfdone, Dd, Ddp, De, Dg, Ddef and Drbrep to FALSE;
- set Denb, Denbb and Depnb to zero;
- set Dps and Dx to FALSE.
- If Du:
  - set Dbcr to FALSE.

Reset variables used for ready flow:

- set Deei to FALSE;
- set Dopi to FALSE;
- set Drdyi to FALSE;
- set Droi to FALSE;
- set D2pc to FALSE;
- set Dcoor to FALSE.

[REWRLOG] (**rewr**iting intermediate **log**)

This procedure is used by a node to replace the log-ready record with a log-commit record [after having entered the DECIDED (commit) state]. After a successful write operation, the dialogue state machine for the dialogue towards the commit coordinator is informed that commit confirmation is now possible.

- If ^Nclw
  - If ^Ldfail:
    - set Snd(Naaid, Nbrid).type to "log-commit";
    - set Nclw to TRUE;
    - generate "log-rewritten".

[SAVEAAIDN] (**save** the **a**tomic **a**ction **id**entifier for the **n**ext transaction)

The atomic action identifier and the branch identifier received with the AF-NOCHANGE (result-requested) indication (on the chaining superior dialogue) are saved.

NOTE 17 – Procedure ADDNODE is invoked if the C-BEGIN indication is received on a chaining dialogue with a subordinate or if a C-BEGIN request is issued on the chaining dialogue with the superior.

- Set Naaidn to the atomic-action-identifier parameter;
- set Nbridn to the atomic-action-branch-identifier parameter.

[SAVECR] (**save** the **c**ompletion **r**eport parameters of the TP-DONE request)

- Set Ncrsev to the severity parameter;
- set Ncrud to the completion-data parameter.

[SEND1PC] [**send** AF/**C-NOCHANGE (result-requested) req**uest]

- If ^Dsup and Dch

  NOTE 18 – This is the static one-phase dialogue with a subordinate, the AF-NOCHANGE (result-requested) req will carry the value of Naaidn (this is the root of the transaction tree).

  - Set Naaidn to a new unique value;
  - set Dbridn to a new unique value;
  - set the atomic-action-identifier parameter to Naaidn;
  - set the atomic-action-branch-identifier to Dbridn;
  - issue a AF-NOCHANGE (result-requested) req.
- If Dsup or ^Dch:
  - set the atomic-action-identifier parameter to NULL;
  - set the atomic-action-branch-identifier to NULL;
  - issue a C-NOCHANGE (result-requested) req.
- Set Dcoor to TRUE.

[SEND2PC] (**send C-READY req**uest)

- Set the TPPM bound data to the ready-to-commit state;
- set Snd(Naaid,Nbrid).type to 'log-ready'.
- If Ptok:
  - issue P-TOKEN-GIVE req.
- If Dsup:
  - set Snd(Naaid,Nbrid).spbr.readyk to ready-sent.
- If ^Dsup:
  - set Snd(Naaid,Nbrid).sbbr(Dbrid).readyk to ready-sent.
- Issue C-READY req.
- Set Dcoor to TRUE.

[SENDEE] (**send** AF-**E**ARLY-**E**XIT request)

- If Dcdfu:
  - set severity parameter to Ncrsev;
  - set completion-data paramater to Ncrud.
- Issue AF-EARLY-EXIT req.
- Set Dcoor to TRUE.

[SENDRDY?] (**send *ready*** *signal / ready-substitute indication**?***)

This procedure is evaluated only once in the termination phase of a transaction. There is exactly one branch without a *ready signal* or a *ready-substitute indication* and *transaction completion request* has already been received and processed by all state machines. The state-machine is in state 20.1 (no *ready signal* and no *ready-substitute indication* received), all other state-machines are in state 20.2 (*ready signal* or *ready-substitute indication* received).

One of the following events is generated in this procedure:

- generate 'enter-ready/sub-state' synchronizing event, if a *ready-signal* or *ready-substitute request* corresponding to the node state is sendable;
- initiate rollback, if the branch is exclusive and no *ready signal* or *ready-substitute request* is sendable;
- send a prepare-message if no *ready signal* or *ready-substitute request* is sent and no rollback is initiated.

If the *ready signal or ready-substitute request* is not sendable on the dialogue by a static constraint and this is no *exclusive branch*, no action (except sending a prepare message) is performed in this procedure. The state machine is still in state 20.1 waiting for a *ready signal* or *ready-substitute indication* from the neighbour.

The local decision Ldready is evaluated when the READY state is to be entered and ready must not be sent, i.e. the branch is no *exclusive branch*.

The procedure is not used if TP-EARLY-EXIT request has been received and there was an early-exit ind from each subordinate (there is exactly one dialogue without a *ready-signal* or a *ready-substitute indication* – the superior dialogue).

If this is the superior dialogue and *the coordination level is one-phase commitment*, enter-ready-state or enter-one-phase-state is not possible but enter-read-only-state may be possible.

```
– If (Nrdyi > 0) or (LDbd or Ncmtr)
  READY state needed
  – If Drsen
    – If Do
      ready signal is sendable.
      – If Dex
        Branch is exclusive, ready signal must be sent:
        – generate 'enter-ready-state'.
      – If ^Dex
        Branch is not exclusive (ready signal or ready-substitute indication is receivable).
        – If ^Ldready
          ready signal is not sent by local decision.
        – If ^Dps:
          – invoke [GENPREP].
        – Exit this procedure.
        ready signal is sent by local decision:
        – generate 'enter-ready-state'.
  – If ^Do
    ready signal is not sendable.
    – If Dex
      This is a static one-phase exclusive subordinate branch.
      A ready signal is not sendable, deadlock, initiate rollback:
      – generate 'rollback-by-tppm'.
    – If ^Dex
      – If ^Dps:
        – invoke [GENPREP].
```

  - If ^Drsen

    ready signal is not sendable and consequently this is not an *exclusive branch* and no rollback is initiated.

    - If ^Dps:

      - invoke [GENPREP].

  - Exit this procedure.

- If (Nopi > 0 or Nopr)

  ONE-PHASE state wanted.

  - If Drsen

    ready-substitute request or ready-signal is sendable.

    - If Dopfu

      ready-substitute request is sendable and sent (there is no local decision not to send):

      - generate 'enter-one-phase-state'.

    - If ^Dopfu

      ready signal is sendable.

      - If Dex

        Branch is exclusive, *ready signal* must be sent, *ready signal* is sendable and sent:

        - generate 'enter-ready-state'.

      - If ^Dex

        Branch is not exclusive (*ready signal* or *ready-substitute indication* is receivable).

        - If ^Ldready

          ready signal and *ready-substitute request* is not sent by local decision.

          - If ^Dps:

            - invoke [GENPREP].

          - Exit this procedure.

        ready signal is sent by local decision:

        - generate 'enter-ready-state'.

  - If ^Drsen

    ready signal or *ready-substitute* request is not sendable and consequently this is not an *exclusive branch* and no rollback is initiated.

    - If ^Dps:

      - invoke [GENPREP].

  - Exit this procedure.

The node and the subtree is read-only, a C-NOCHANGE (result-not-required) request is issued.

The read-only f.u. must be selected on the superior dialogue (service constraint) and this is the superior dialogue (otherwise C-READY indication or C-NOCHANGE (result-requested) indication must have been received on the superior dialogue and the node is not read-only).

- Generate 'enter-read-only-state'.

[SENDREP?] (**send rep**ort to superior if possible – Without abort)

This procedure is invoked after receiving the sync. event 'send-report' on the dialogue with the superior and in state 21.5.1. and during recovery on the superior dialogue. The reporting status is known at the node and reporting is necessary.

NOTE 19 – If there is recovery in progress and no channel is attached, the report is not sendable in this procedure.

- If ^Db
  - If Dcdfu:
    - set the severity parameter to Ncrsev;
    - set the completion-data paramater to Ncrud.
  - If ^Dhrsfu
    - If memsp (SldD, Naaid, Nbrid):
      - set *heuristic-report* parameter to SldD (Naaid, Nbrid).type.
    - If ^memsp (SldD, Naaid, Nbrid):
      - set *heuristic-report* parameter to 'none'.
  - Issue AF-REPORT(dataRI) req.
- If Db and Dchat and ^Dhrsfu

  Heuristic reporting and a channel is attached, never completion reporting, issue report on the channel.

  - If memsp (SldD, Naaid, Nbrid):
    - set *heuristic-report* parameter to SldD (Naaid, Nbrid).type.
  - If ^memsp (SldD, Naaid, Nbrid):
    - set *heuristic-report* parameter to 'none'.
  - Issue AF-REPORT(commitRI) req.

NOTE 20 – If the dialogue is aborted and no channel is attached, a CAF-GIVE ind (after CAF-PLEASE req) or a channel from the superior may follow. If the channel is attached and Nhrst is true and the first TP-DONE req has been received, the report is sendable on the channel (on data if the node is on the OCC-path or otherwise with the commit confirm).

[SENDREP?AB] (**send rep**ort to superior if possible – With **ab**ort)

This procedure is invoked after receiving the sync. event 'send-report' on the dialogue with the superior and in state 21.5.1. The reporting status is known at the node and reporting is necessary and a TP-U-ABORT request is pending and now sendable; detach of the dialogue is performed in the calling cell.

- If ^Db
  - If Dcdfu:
    - set the severity parameter to Ncrsev;
    - set the completion-data parameter to Ncrud.
  - If ^Dhrsfu
    - If memsp (SldD, Naaid, Nbrid) and ^Dhrsfu:
      - set *heuristic-report* parameter to SldD (Naaid, Nbrid).type.
    - If ^memsp (SldD, Naaid, Nbrid):
      - set *heuristic-report* parameter to 'none'.
  - Issue AF-ABORT-AND-REPORT (dataRI) req.
  - If Dtb and Du:
    - set Dtb to FALSE;
    - set Dbpart to TRUE.

[SENDRO] [**send C-NOCHANGE (result-not-required) req**uest]

- Issue a C-NOCHANGE (result-not-required) req;
- set Dcoor to TRUE.

[SETAAID] (**set** TPPM **a**tomic-**a**ction, atomic-action-branch, and superior **id**entifiers)

Sets the *atomic-action-identifier*, *atomic-action-branch-identifier*, and *superior* required to find the TPPM which requested the channel.

- Set Caaid to the *atomic-action-identifier* parameter;
- set Cbrid to the *atomic-action-branch-identifier* parameter;
- set Csup to the *superior* parameter.

[SETDIAG] (**set diag**nostic)

Sets the *diagnostic* parameter of the next AF- or TP-service primitive issued by the TPPM.

- If this is a protocol error:
  - set the *diagnostic* parameter to "protocol-error".
- If this is an internal error
  - If Ldperm:
    - set the *diagnostic* parameter to "permanent-failure";
    - exit this procedure.
- Set the *diagnostic* parameter to "transient-failure".

[SETDIAGBD] (**set diag**nostic on AF-**B**EGIN-**D**IALOGUE response)

- If the Dialogue functional unit is selected, set the *diagnostic* parameter to, as appropriate, one of:
  - "recipient-tpsu-title-unknown";
  - "tpsu-not-available (permanent)";
  - "tpsu-not-available (transient)";
  - "recipient-tpsu-title-required";
  - "functional-unit-not-supported";
  - "functional-unit-combination-not-supported";
  - "no-reason-given".
- If the Recovery functional unit is selected, set the *diagnostic* parameter to, as appropriate, one of:
  - "functional-unit-not-supported";
  - "tppm-recovery-not-available";
  - "two-way-recovery-not-supported";
  - "no-reason-given".
- If the *diagnostic* parameter is set to "functional-unit-not-supported":
  - set the *functional-units* parameter to the functional units that are supported.

[SETDIAGEC] (**set diag**nostic – **e**arly-exit-transaction-**c**ompletion-collision)

Sets the *diagnostic* parameter of the next AF- or TP- service primitive issued by the TPPM.

- Set the *diagnostic* parameter to "early-exit-transaction-completion-collision".

[SETDIAGLO] (**set diag**nostic – **lo**cal-rollback)

Sets the *diagnostic* parameter of the next AF- or TP- service primitive issued by the TPPM.

- Set the *diagnostic* parameter to "local-rollback".

[SETDIAGSB] (**set diag**nostic – **sub**ordinate-rollback)

Sets the *diagnostic* parameter of the next AF- or TP-service primitive issued by the TPPM.

- Set the *diagnostic* parameter to "subordinate-rollback".

[SETDIAGSP] (**set diag**nostic – **sup**erior-rollback)

Sets the *diagnostic* parameter of the next AF- or TP-service primitive issued by the TPPM.

- Set the *diagnostic* parameter to "superior-rollback".

[SETDIAGTP] (**set diag**nostic on **T**P-**P**-ABORT-indication)

- If this is a Protocol error:
  - set the *diagnostic* parameter to "protocol-error".
- If this is an Internal error
  - If Ldperm:
    - set the *diagnostic* parameter to "permanent-failure";
    - exit this procedure.
  - Set the *diagnostic* parameter to "transient-failure".
- If this is an A-RELEASE rsp or A-RELEASE cnf:
  - set the *diagnostic* parameter to "permanent-failure".
- If this is an A-ABORT ind:
  - set the *diagnostic* parameter to "permanent-failure".
- If this is an A-P-ABORT ind:
  - set the *diagnostic* parameter to "permanent-failure".
- If this is an A-ABORT req:
  - set the *diagnostic* parameter to "permanent-failure".
- If this is a CAF-RECOVER (ready) ind or a CAF-RECOVER (commit) ind:
  - set the *diagnostic* parameter to "permanent-failure".

[SETDIAGUC] (**set diag**nostic – **u**ser-data-transaction-**c**ompletion-collision)

Sets the *diagnostic* parameter of the next AF- or TP-service primitive issued by the TPPM.

- Set the *diagnostic* parameter to "user-data-transaction-completion-collision".

[SETTOKX] (**set** A**tokx** to TRUE)

Sets Atokx to TRUE after a C-RECOVER indication or a CAF-RECOVER indication has been received on a two-way recovery channel.

- If Atwr:
  - set Atokx to TRUE;
  - set Ctokr to FALSE.

[SNDORDCD] [send a NOCHANGE request or make a decision (forced collision or decision)]

This procedure is invoked if one C-READY indication was received and there are no bound data at the node.

This procedure is invoked by the state machine by which the one C-READY indication was received. If the node is able to send a *ready-substitute request*, this will be a C-NOCHANGE (result-not-required) request if the read-only f.u. is selected and this is the dialogue to the superior (and some further conditions). Otherwise the resulting event is a C-NOCHANGE (result-requested) request. If the node is not able to send a *ready-substitute request*, the procedure [DECISION] is invoked.

- If Dsup and Drofu and Nopi = 0 and Nror:
  - generate 'enter-read-only-state';
  - exit this procedure.
- If Dopfu and Drsen and (Nror or Nopr):
  - generate 'enter-one-phase-state';
  - exit this procedure.

If no *ready-substitute request* has been sent, then node is the commit coordinator.

- Invoke [DECISION].

[TRBi] (issue **T**P-**R**OLL**B**ACK **i**ndication)

The diagnostic parameter is optional when the Completion Diagnostics functional unit is selected on any *coordinated dialogue (i.e. Ncntcdfu* > *0)* and is absent otherwise.

- If Ncntcdfu = 0:
    - issue TP-ROLLBACK ind without diagnostic parameter.
- If Ncntcdfu > 0:
    - issue TP-ROLLBACK ind with diagnostic parameter.

[TREERESET] (**reset tree** extension variables)

Upon deletion of a branch during active phase of the transaction and during completion of a transaction, the variables used for the checking of the tree extension rules are reset according to the type of the deleted branch. This procedure is called in the subcells containing [DELBRANCH] and in the procedures completing a transaction at the node.

- If Dex:
    - set Dex to FALSE;
    - set Nex to FALSE.
- If Dsopex:
    - set Dsopex to FALSE;
    - set Nsopex to FALSE.
- If D2exp:
    - set D2exp to FALSE;
    - subtract one unit from Ncnt2exp.
- If Ncnt2exp = 0:
    - set N2exp to FALSE.

Reset dialogue types, which are only valid if the check-ready-directions parameter is specified (either absent or TRUE or FALSE).

- If Dsup
    - If Dgrp = 1 or Dgrp = 3:
        - set Dgrp to 2.
- If ^Dsup
    - If Dgrp = 6 or Dgrp = 8:
        - set Dgrp to 7.
    - If Dgrp = 10 or Dgrp = 12:
        - set Dgrp to 11.

[TREESET] (**set tree** extension variables)

Upon creation of a new branch at the node, the variables used for checking of the tree extension rules are set according to the type of the new branch.

- If Dsup

    The superior branch is the first branch at the node (if any).

    - If Dgrp ^= 3
        - If Ldtch:
            - set Ntch to TRUE.
        - If Dgrp = 3

            Tree checking is ordered by the superior:

            - set Ntch to TRUE.

- If ^Dsup and ^Ntch and Naaidn = NULL

  The node is root and the subordinate branch is the first branch at the node.

  - If Dgrp ^= 8 and Dgrp ^= 12

    With type 8 and type 12, tree checking is not allowed at the node (the subordinate is not ordered to check tree extension).

    - If Ldtch:
      - set Ntch to TRUE.
- If Dsup
  - If Dgrp = 1 and Ntch:
    - set Dex to TRUE;
    - set Nex to TRUE.
  - If Dgrp = 3:
    - set Dex to TRUE;
    - set Nex to TRUE.
- if ^Dsup
  - If Dgrp = 1 and Ntch:
    - set Dsopex to TRUE;
    - set Nsopex to TRUE;
    - set Dex to TRUE;
    - set Nex to TRUE.
  - If Dgrp = 2:
    - set Dsopex to TRUE;
    - set Nsopex to TRUE;
    - set Dex to TRUE;
    - set Nex to TRUE.
  - If Dgrp = 3 and Ntch:
    - set Dex to TRUE;
    - set Nex to TRUE.
  - If Dgrp = 4:
    - set Dex to TRUE;
    - set Nex to TRUE.
  - If Dgrp = 5 and Ntch:
    - set D2exp to TRUE;
    - add one unit to Ncnt2exp;
    - set N2exp to TRUE.
  - If Dgrp = 6 and Ntch:
    - set D2exp to TRUE;
    - add one unit to Ncnt2exp;
    - set N2exp to TRUE.
  - If Dgrp = 9 or Dgrp = 10 or Dgrp = 12:
    - set D2exp to TRUE;
    - add one unit to Ncnt2exp;
    - set N2exp to TRUE.

[TREP] (issue **rep**ort indication to **T**PSUI)

- If ^Dhrsfu and heuristic-report parameter ^= NONE:
  - issue TP-HEURISTIC-REPORT ind.
- If Dcdfu and (severity parameter ^= EMPTY or completion-data parameter ^= EMPTY):
  - issue TP-COMPLETION-REPORT ind.

  Completion **r**eport parameters **all**owed with next TP-DONE req?

  and make a TP-DONE req owed?

  - If ^Nr and Ntbicr and ^Dcrpa:
    - generate 'cr-allowed'.
  - If ^Dd:
    - generate "Set-done-true".
    - If ^Drbrep
      - If Nr or Ni, add Nsubnb to Ncnt.
      - If Ni or Nlf, add one unit to Ncnt.

A.4.4.5 Actions after node crash

After a node crash, action REBUILDTPPMS is executed.

[REBUILDTPPMS] (**Rebuild TPPMs** after node crash)

Creates a TPPM if one does not exist for each element of SnD having a non NULL type field after a node crash has occurred.

- For each element of SnD where type ^= NULL and no TPPM exists for the node:
  - create a new MACF and for this new MACF do:
    - set Naaid to aaid;
    - set Nbrid to spbr.brid;
    - set Nt to TRUE.

  Create branch to superior:

    - If Nbrid ^= NULL and spbr.readyk = ready-sent

      NOTE 1 – The superior branch is the branch towards the coordinator.

      - Add a new state machine and for this new branch do:
        - set Dsup to TRUE;
        - set Dcoor to TRUE;
        - set Db, Dl, and Danyb to TRUE.
          - If spbr.hreport = none:
            - set Dhrsfu to TRUE.
          - If type = "log-commit":
            - set Dd to TRUE;
            - set D2pc to TRUE;
            - set state to 21.5.
          - If type = "log-ready":
            - set state to 20.3.
    - If Nbrid ^= NULL and spbr.readyk = ready-received

      NOTE 2 – The superior branch is **not** the branch towards the coordinator.

- Add a new state machine and for this new branch do:
  - set Dsup to TRUE;
  - set Drdyi to TRUE;

    NOTE 3 – The node is the coordinator if there is no branch to a subordinate with readyk set to ready-sent.
  - set Db, Dl, and Danyb to TRUE.
  - If spbr.hreport = none:
    - set Dhrsfu to TRUE.
  - If type = "log-commit" and ^Dhrsfu

    Heuristic reporting applies on the channel with the superior:
    - set Ncnt to 1;
    - set Dd to TRUE;
    - set D2pc to TRUE;
    - set state to 21.5.1.
  - If type = "log-commit" and Dhrsfu

    No heuristic reporting on the channel with the superior:
    - set Ncnt to 1;
    - set Dd to TRUE;
    - set D2pc to TRUE;
    - set state to 21.5.2.
  - If type = "log-ready":
    - set state to 20.3.

Create branches to subordinates:

- for each element of sbbr.
  - If readyk = ready-received:
    - add one unit to Nsubnb;
    - add one unit to Ncnt

      (commit confirm awaited);
    - add a new state machine and for this new branch do:
      - set Dbrid to brid;
      - set Db, Dl, and Danyb to TRUE;
      - set Drdyi to TRUE.
      - If sbbr(Dbrid).hreport = none:
        - set Dhrsfu to TRUE.
      - If type = "log-commit":
        - set Dd to TRUE;
        - set state to 21.1.
      - If type = "log-ready":
        - set state to 20.3.
  - If readyk = ready-sent:

    NOTE 4 – The node is not the coordinator.
    - set Dcoor to TRUE;
    - add one unit to Nsubnb;
    - add a new state machine and for this new branch do:
      - set Dbrid to brid;
      - set Db, Dl, and Danyb to TRUE.

```
                    – If sbbr(Dbrid).hreport = none:
                        – set Dhrsfu to TRUE.
                    – If type = "log-commit" and Dhrsfu:
                        – set Dd to TRUE;
                        – set state to 21.3.
                    – If type = "log-commit" and ^Dhrsfu:
                        – add one unit to Ncnt
                          (heuristic report awaited);
                        – set Dd to TRUE;
                        – set state to 21.1.
                    – If type = "log-ready":
                        – set state to 20.3.
– If Nbrid = NULL:
    – set Nr to TRUE.
– If Nbrid ^= NULL and Nsubnb = 0:
    – set Nlf to TRUE.
– If Nbrid ^= NULL and Nsubnb ^= 0:
    – set Ni to TRUE.
  Add a new node finite state machine and for this new state machine do:
    – set Dl to TRUE;
    – set state to 26.2.
    – If type = "log-commit":
        – set state to 26.3;
        – set Dd to TRUE;
        – add one unit to Ncnt.
    – If type = "log-commit"
      Count TP-DONE requests (like OWEDONE).
        – If Nr or Ni:
            – add Nsubnb to Ncnt.
        – If Ni or Nlf:
            – add one unit to Ncnt.
        – Issue a TP-COMMIT ind.
– Generate "Restart-TPPM".
```

A.5 TPASE

A.5.1 TPASE states

There is no state defined for the TPASE.

A.5.2 TPASE variables

There is no variable defined for the TPASE.

A.5.3 TPASE events

There is no Internal event and no synchronizing event defined for the TPASE .

A.5.4 TPASE actions

The following actions are defined for the TPASE:

Dec

– decode the TP APDU;

– issue the AF service primitive corresponding to the received TP APDU to the SACF.

MapA (**Map** to **A-**ABORT service)

– encode the TP APDU;

– issue an A-ABORT request, with the TP APDU carried by the *user data* parameter.

MapC (**Map** to **C**CR)

– encode the TP APDU;

– issue the CCR request/response specified by the *mapping* parameter, with the TP APDU carried by the *user data* parameter.

MapPd (**Map** to **P-D**ATA service)

– encode the TP APDU;

– issue a P-DATA request, with the TP APDU carried by the *user data* parameter.

NOTE – The actual Presentation service which will be used to carry the APDU is determined according to the rules contained in 9.5 "Mapping" and 10.7 "Concatenation".

MapPp (**Map** to **P**-TOKEN-**P**LEASE service)

– encode the TP APDU;

– issue a P-TOKEN-PLEASE request, with the TP APDU carried by the *user data* parameter.

MapPg (**Map** to **P**-TOKEN-**G**IVE service)

– encode the TP APDU;

– issue a P-TOKEN-GIVE request, with the TP APDU carried by the *user data* parameter.

MapS (**Map** to **S**ACF)

– decode the TP APDU carried by the *user data* parameter;

– set the *mapping* parameter to the value corresponding to the received CCR indication/confirm or A-ABORT indication;

– issue the AF service primitive corresponding to the received TP APDU to the SACF.

Table A.11 lists the actions taken for each event received by the TPASE.

A.6 SACF

A.6.1 SACF states

The names for SACF states in the main text are shown after the corresponding state number.

State 1 (FREE)

The SAO is in the FREE state.

State 1.1

This state is valid only for a contention-winner. An AF-BID (*token-requested* = TRUE) indication has been received. The *token* is not available and the SACF waits for the *token* before accepting or rejecting the bid.

Table A.11/X.862 – TP-ASE actions *(sheet 1 of 2)*

Event	Action
AF-BEGIN-DIALOGUE req	MapPd
TP-BEGIN-DIALOGUE-RI	Dec
AF-BEGIN-DIALOGUE rsp	MapPd
TP-BEGIN-DIALOGUE-RC	Dec
AF-SOLICIT-DIALOGUE (tokengiveRI) req	MapPg
TP-SOLICIT-DIALOGUE-RI	Dec
C-ROLLBACK (TP-BEGIN-DIALOGUE-RC) cnf	MapS
AF-BID req	MapPd
TP-BID-RI	Dec
AF-BID rsp	MapPd
TP-BID-RC	Dec
AF-END-DIALOGUE req	MapPd
TP-END-DIALOGUE-RI	Dec
AF-END-DIALOGUE rsp	MapPd
TP-END-DIALOGUE-RC	Dec
AF-U-ERROR req	MapPd
TP-U-ERROR-RI	Dec
AF-U-ERROR rsp	MapPd
TP-U-ERROR-RC	Dec
AF-ABORT (user, dataRI) req	MapPd
TP-ABORT-RI	Dec
AF-REPORT (dataRI) req	MapPd
TP-REPORT-RI	Dec
AF-ABORT-AND-REPORT (dataRI) req	MapPd
TP-ABORT+TP-REPORT-RI	Dec
AF-ABORT (provider, abortRI) req	MapA
A-ABORT (TP-ABORT-RI) ind	MapS
AF-GRANT-CONTROL req	MapPd
TP-GRANT-CONTROL-RI	Dec
AF-REQUEST-CONTROL req	MapPd
TP-REQUEST-CONTROL-RI	Dec
AF-HANDSHAKE req	MapPd
TP-HANDSHAKE-RI	Dec
AF-HANDSHAKE rsp	MapPd
TP-HANDSHAKE-RC	Dec
AF-HANDSHAKE-AND-GRANT-CONTROL req	MapPd
TP-HANDSHAKE-AND-GRANT-CONTROL-RI	Dec
AF-HANDSHAKE-AND-GRANT-CONTROL rsp	MapPd
TP-HANDSHAKE-AND-GRANT-CONTROL-RC	Dec
AF-DEFER req	MapPd
TP-DEFER-RI	Dec
AF-PREPARE req	MapC
C-PREPARE (TP-PREPARE-RI) ind	MapS
AF-ABORT (user, commitRI) req	MapC
C-NOCHANGE (TP-ABORT-RI) ind	MapS
C-NOCHANGE (TP-ABORT-RI) cnf	MapS
AF-ABORT (nochangeRC) req	MapC
C-COMMIT (TP-ABORT-RI) ind	MapS
AF-ABORT (user, commitRC) req	MapC
C-COMMIT (TP-ABORT-RI) cnf	MapS
AF-REPORT (commitRC) req	MapC
AF-REPORT (commitRI) req	MapC

Table A.11/X.862 – TP-ASE actions *(sheet 2 of 2)*

Event	Action
C-COMMIT (TP-REPORT-RI) cnf	MapS
AF-ABORT-AND-REPORT (commitRC) req	MapC
C-COMMIT (TP-ABORT-RI, TP-REPORT-RI) cnf	MapS
AF-ABORT (user, rollbackRI) req	MapC
C-ROLLBACK (TP-ABORT-RI) ind	MapS
AF-REPORT (rollbackRI) req	MapC
C-ROLLBACK (TP-REPORT-RI) ind	MapS
AF-ABORT-AND-REPORT (rollbackRI) req	MapC
AF-EARLY-EXIT req	MapC
C-ROLLBACK (TP-ABORT-RI, TP-REPORT-RI) ind	MapS
C-ROLLBACK (TP-EARLY-EXIT-RI) ind	MapS
AF-ABORT (user, rollbackRC) req	MapC
C-ROLLBACK (TP-ABORT-RI) cnf	MapS
AF-HEURISTIC-REPORT (rollbackRC) req	MapC
C-ROLLBACK (TP-REPORT-RI) cnf	MapS
AF-ABORT-AND-REPORT (rollbackRC) req	MapC
C-ROLLBACK (TP-ABORT-RI, TP-REPORT-RI) cnf	MapS
AF-TOKEN-GIVE req	MapPg
TP-TOKEN-GIVE-RI	Dec
AF-TOKEN-PLEASE req	MapPp
TP-TOKEN-PLEASE-RI	Dec

State 1.2

This state is valid only for a contention-winner. An AF-BID indication has been received, and the TPPM accepted the bid. The SACF waits for an AF-BEGIN-DIALOGUE indication from the contention-loser.

State 2 (STRAY)

AF-BEGIN-DIALOGUE request has been issued.

State 3 (BIDDING)

An AF-BID request has been issued.

State 4 (BID CONFIRM RECEIVED)

An AF-BID (accepted) confirm has been received.

State 6 (BUSY)

One of the following situations has occurred:

a) an AF-BEGIN-DIALOGUE indication has been received by a contention-loser, or has been received by a contention-winner while no AF-BEGIN-DIALOGUE request has been issued;

b) an AF-BEGIN-DIALOGUE confirm with a valid *correlator* parameter has been received; or

c) a C-ROLLBACK indication or confirm has been received.

State 7 (CLEANUP ROLLBACK INDICATION EXPECTED)

A superior has issued a C-BEGIN request and received a non-confirmed dialogue termination AF indication.

State 8 (CLEANUP BEGIN INDICATION EXPECTED)

A subordinate provider has rejected a dialogue with coordination level COMMITMENT.

State 9 (CLEANUP ROLLBACK CONFIRM EXPECTED)

A superior has issued a C-ROLLBACK request and received a non-confirmed dialogue termination AF indication.

A.6.2 SACF variables

A.6.2.1 Overview

Two categories of variables are defined for the SACF:

a) variables that pertain to an association. These variables are created at association establishment time, and are destroyed at association termination time. Association variables are prefixed by the letter "A". They are listed in Table A.12; and

b) variables that model a decision local to the node, when there is a choice. Local decision variables are prefixed by the letter "L". They are listed in Table A.19.

NOTE – An important subcategory are the A-variables shared with the MACF [see A.4.2.1 f)].

Table A.12/X.862 – Association variables

Name	Meaning
Aaet	peer **AE**-**t**itle
Abm	**b**id **m**andatory
Abtr	**b**egin-**t**ransaction-**r**eject
Acbegq	**C**-**BEG**IN re**q**uest received
Acopy	**copy** issued AF- service
Adc	**d**ialogue **c**orrelator
Adru	**d**ialogue **r**ejected by **u**ser
Adt	SAF-**D**E**T**ACH-ASSOCIATION request received
Af	C-BEGIN **f**ear
Alpi	**l**ast **p**artner **i**dentifier
Anfd	**n**ot the **f**irst **d**ialogue on association
Aq	**q**ueue
Arrh	**r**eceived **r**ecovery-context-**h**andle
Arvyrs	**r**eco**v**er**y** **res**ponse awaited
Atokr	**tok**en **r**equested
Atokx	**tok**en e**x**pected (two-way recovery only)
Atppm	attached to a **TPPM**
Atwr	**t**wo-**w**ay-**r**ecovery
Aw	contention-**w**inner

Table A.13/X.862 – Dialogue (sheet 1 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates Event		Dsup		^Dsh		^Dsh
TP-BEGIN-DIALOGUE (Group 1, transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VDgrp1] [TREESET] [ADDBRSB] [VNtpsuiT] [VAtppmT] [ABDrq] [CBErq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr ^Nt ^Ntch Nr [INITDIASB] [VDgrp1] [TREESET] [ADDBRSB] [VAtppmT] [ABDrq] [CBErq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr ^Nt ^Nr, ^Ni, ^Nlf [INITDIASB] [VDgrp1] [TREESET] [ADDBRSB] [VAtppmT] [ABDrq] [CBErq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					

Table A.13/X.862 (sheet 2 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMIT req or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 3 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates Event		Dsup		^Dsh		^Dsh
TP-BEGIN-DIALOGUE (Group 2, transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VDgrp2] [TREESET] [ADDBRSB] [VNtpsuiT] [VAtppmT] [ABDrq] [CBErq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr ^Nt ^Nex, ^N2exp Nr [INITDIASB] [VDgrp2] [TREESET] [ADDBRSB] [VAtppmT] [ABDrq] [CBErq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr ^Nt ^Nr, ^Ni, ^Nlf [INITDIASB] [VDgrp2] [TREESET] [ADDBRSB] [VAtppmT] [ABDrq] [CBErq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					

Table A.13/X.862 (sheet 4 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 5 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-BEGIN-DIALOGUE (Group 3, transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VDgrp3] [TREESET] [ADDBRSB] [VNtpsuiT] [VAtppmT] [ABDrq] [CBErq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr ^Nt ^Ntch [INITDIASB] [VDgrp3] [TREESET] [ADDBRSB] [VAtppmT] [ABDrq] [CBErq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr ^Nt Ntch ^Nex [INITDIASB] [VDgrp3] [TREESET] [ADDBRSB] [VAtppmT] [ABDrq] [CBErq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					

Table A.13/X.862 (sheet 6 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 7 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates Event		Dsup		^Dsh		^Dsh
TP-BEGIN-DIALOGUE (Group 4, transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VDgrp4] [TREESET] [ADDBRSB] [VNtpsuiT] [VAtppmT] [ABDrq] [CBErq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr ^Nt ^Nex [INITDIASB] [VDgrp4] [TREESET] [ADDBRSB] [VAtppmT] [ABDrq] [CBErq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					
TP-BEGIN-DIALOGUE (Group 5, transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VDgrp5] [TREESET] [ADDBRSB] [VNtpsuiT] [VAtppmT] [ABDrq] [CBErq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr ^Nt ^Ntch [INITDIASB] [VDgrp5] [TREESET] [ADDBRSB] [VAtppmT] [ABDrq] [CBErq] 2					

(Continued on sheet 9 of 98)

Table A.13/X.862 (sheet 8 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 9 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates Event		Dsup		^Dsh		^Dsh
TP-BEGIN-DIALOGUE (Group 5, transaction branch) req *(Concluded 2 of 2)*	^Ldrej, ^Pnew Ntpsui, ^Ncr ^Nt Ntch ^Nsopex [INITDIASB] [VDgrp5] [TREESET] [ADDBRSB] [VAtppmT] [ABDrq] [CBErq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					
TP-BEGIN-DIALOGUE (Group 6, transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VDgrp6] [TREESET] [ADDBRSB] [VNtpsuiT] [VAtppmT] [ABDrq] [CBErq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr ^Nt ^Ntch [INITDIASB] [VDgrp6] [TREESET] [ADDBRSB] [VAtppmT] [ABDrq] [CBErq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr ^Nt Ntch ^Nsopex [INITDIASB] [VDgrp6] [TREESET] [ADDBRSB] [VAtppmT] [ABDrq] [CBErq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					

Table A.13/X.862 (sheet 10 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 11 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-BEGIN-DIALOGUE (Group 8, transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VDgrp8] [TREESET] [ADDBRSB] [VNtpsuiT] [VAtppmT] [ABDrq] [CBErq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr ^Nt ^Ntch [INITDIASB] [VDgrp8] [TREESET] [ADDBRSB] [VAtppmT] [ABDrq] [CBErq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					
TP-BEGIN-DIALOGUE (Group 9, transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VDgrp9] [TREESET] [ADDBRSB] [VNtpsuiT] [VAtppmT] [ABDrq] [CBErq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr ^Nt ^Nsopex [INITDIASB] [VDgrp9] [TREESET] [ADDBRSB] [VAtppmT] [ABDrq] [CBErq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					

Table A.13/X.862 (sheet 12 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 13 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-BEGIN-DIALOGUE (Group 10, transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VDgrp10] [TREESET] [ADDBRSB] [VNtpsuiT] [VAtppmT] [ABDrq] [CBErq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr ^Nt ^Nsopex [INITDIASB] [VDgrp10] [TREESET] [ADDBRSB] [VAtppmT] [ABDrq] [CBErq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					
TP-BEGIN-DIALOGUE (Group 12, transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VDgrp12] [TREESET] [ADDBRSB] [VNtpsuiT] [VAtppmT] [ABDrq] [CBErq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr ^Nt, ^Ntch ^Nsopex [INITDIASB] [VDgrp12] [TREESET] [ADDBRSB] [VAtppmT] [ABDrq] [CBErq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					

Table A.13/X.862 (sheet 14 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 15 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-BEGIN-DIALOGUE (Group 1, no transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp1] [ABDrq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp1] [ABDrq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					
TP-BEGIN-DIALOGUE (Group 2, no transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp2] [ABDrq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp2] [ABDrq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					

Table A.13/X.862 (sheet 16 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 17 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-BEGIN-DIALOGUE (Group 3, no transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp3] [ABDrq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp3] [ABDrq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					
TP-BEGIN-DIALOGUE (Group 4, no transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp4] [ABDrq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp4] [ABDrq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					

Table A.13/X.862 (sheet 18 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 19 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-BEGIN-DIALOGUE (Group 5, no transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp5] [ABDrq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp5] [ABDrq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					
TP-BEGIN-DIALOGUE (Group 7, no transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp7] [ABDrq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp7] [ABDrq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					

Table A.13/X.862 (sheet 20 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 21 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-BEGIN-DIALOGUE (Group 9, no transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp9] [ABDrq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp9] [ABDrq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					
TP-BEGIN-DIALOGUE (Group 11, no transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp11] [ABDrq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp11] [ABDrq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					

Table A.13/X.862 (sheet 22 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 23 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-BEGIN-DIALOGUE (Group 13, no transaction branch) req	^Ldrej, Pnew [INITMACF] [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp13] [ABDrq] 2					
	^Ldrej, ^Pnew Ntpsui, ^Ncr [INITDIASB] [VNtpsuiT] [VAtppmT] [VDgrp13] [ABDrq] 2					
	Ldrej, Ldunk [TBDcRPu] 1					
	Ldrej, ^Ldunk [TBDcRP] 1					
AF-BEGIN-DIALOGUE (Group 2, Polarized Control fu selected, No transaction branch) ind	^Ldrej [INITMACF] [INITDIASP] [VNtpsuiT] [VAtppmT] [VDgrp2] [TBDi] 3					
	Ldrej [SETDIAGBD] [ABDrsRPd] [SDETrqBF] 1					
AF-BEGIN-DIALOGUE (Group 3, Polarized Control fu selected, No transaction branch) ind	[INITDIASP] [VNtpsuiT] [VAtppmT] [VDgrp3] [TBDi] 3					
	Ldrej [SETDIAGBD] [ABDrsRPd] [SDETrqBF] 1					
AF-BEGIN-DIALOGUE (Group 4, Polarized Control fu selected, No transaction branch) ind	[INITDIASP] [VNtpsuiT] [VAtppmT] [VDgrp4] [TBDi] 3					
	Ldrej [SETDIAGBD] [ABDrsRPd] [SDETrqBF] 1					

Table A.13/X.862 (sheet 24 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 25 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates Event		Dsup		^Dsh		^Dsh
AF-BEGIN-DIALOGUE (Group 5, Polarized Control fu selected, No transaction branch) ind	^Ldrej [INITMACF] [INITDIASP] [VNtpsuiT] [VAtppmT] [VDgrp5] [TBDi] 3					
	Ldrej [SETDIAGBD] [ABDrsRPd] [SDETrqF] 1					
AF-BEGIN-DIALOGUE (Group 2, Shared Control fu selected, No transaction branch) ind	^Ldrej [INITMACF] [INITDIASP] [VNtpsuiT] [VAtppmT] [VDgrp2] [TBDi] 2					
	Ldrej [SETDIAGBD] [ABDrsRPd] [SDETrqBF] 1					
AF-BEGIN-DIALOGUE (Group 3, Shared Control fu selected, No transaction branch) ind	^Ldrej [INITMACF] [INITDIASP] [VNtpsuiT] [VAtppmT] [VDgrp3] [TBDi] 2					
	Ldrej [SETDIAGBD] [ABDrsRPd] [SDETrqBF] 1					
AF-BEGIN-DIALOGUE (Group 4, Shared Control fu selected, No transaction branch) ind	^Ldrej [INITMACF] [INITDIASP] [VNtpsuiT] [VAtppmT] [VDgrp4] [TBDi] 2					
	Ldrej [SETDIAGBD] [ABDrsRPd] [SDETrqBF] 1					

Table A.13/X.862 (sheet 26 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 27 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates Event		Dsup		^Dsh		^Dsh
AF-BEGIN-DIALOGUE (Group 5, Shared Control fu selected, No transaction branch) ind	^Ldrej [INITMACF] [INITDIASP] [VNtpsuiT] [VAtppmT] [VDgrp5] [TBDi] 2					
	Ldrej [SETDIAGBD] [ABDrsRPd] [SDETrqF] 1					
AF-BEGIN-DIALOGUE (Group 1, Shared Control fu selected or Polarized Control fu selected, transaction branch) ind	^Ldrej [INITMACF] [INITDIASP] [VAtppmT] [VDbegdiSAVE] [VDgrp1] 1.1					
	Ldrej [SETDIAGBD] [ABDrsRPd] [SDETrqCB] 1					
AF-BEGIN-DIALOGUE (Group 3, Shared Control fu selected or Polarized Control fu selected, transaction branch) ind	^Ldrej [INITMACF] [INITDIASP] [VAtppmT] [VDbegdiSAVE] [VDgrp3] 1.1					
	Ldrej [SETDIAGBD] [ABDrsRPd] [SDETrqCB] 1					
AF-BEGIN-DIALOGUE (Group 4, Shared Control fu selected or Polarized Control fu selected, transaction branch) ind	^Ldrej [INITMACF] [INITDIASP] [VAtppmT] [VDbegdiSAVE] [VDgrp4] 1.1					
	Ldrej [SETDIAGBD] [ABDrsRPd] [SDETrqCB] 1					
TP-BEGIN-DIALOGUE (accepted) rsp			Dsup Ncr [DELIMIT] 2	Dsup Ncr [DELIMIT] 3		Dsup Ncr [DELIMIT] 5

Table A.13/X.862 (sheet 28 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
					Dsup Ncr [DELIMIT] 18		

Table A.13/X.862 (sheet 29 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates Event		Dsup		^Dsh		^Dsh
TP-BEGIN-DIALOGUE (rejected) rsp			^Du, ^Dl, Dsup ^Nrn, ^Da [ABDrsRUd] [SDETrqF] 1	^Du, ^Dl, Dsup ^Nrn, ^Da [ABDrsRUd] [SDETrqF] 1		^Du, ^Dl, Dsup ^Nrn, ^Da [ABDrsRUd] [SDETrqF] 1
			Du, ^Dl, Dsup ^Nrn, ^Da [ABDrsRUd] [SDETrqBF] 1	Du, ^Dl, Dsup ^Nrn, ^Da [ABDrsRUd] [SDETrqBF] 1		Du, ^Dl, Dsup ^Nrn, ^Da [ABDrsRUd] [SDETrqBF] 1
			Dl, Dsup ^Nrn, ^Da [ABDrsRUr] [SDETrqRBC] [REJTRAN] [TREERESET] 1	Dl, Dsup ^Nrn, ^Da [ABDrsRUr] [SDETrqRBC] [REJTRAN] [TREERESET] 1		Dl, Dsup ^Nrn, ^Da [ABDrsRUr] [SDETrqRBC] [REJTRAN] [TREERESET] 1
AF-BEGIN-DIALOGUE (accepted, dataRI) cnf			^Dsup Dcr [TBDcX] [VDcrF] [VDaT] 2	^Dsup Dcr [TBDcX] [VDcrF] [VDaT] 3	^Dsup Dcr [TBDcX] [VDcrF] [VDaT] 4	
			^Dsup ^Dcr [VDaT] 2	^Dsup ^Dcr [VDaT] 3	^Dsup ^Dcr [VDaT] 4	
AF-BEGIN-DIALOGUE (rejected(provider), dataRI) cnf			^Dl, ^Dsup [TBDcX] [SDETrqF] 1	^Dl, ^Dsup [TBDcX] [SDETrqF] 1	^Dl, ^Dsup [TBDcX] [SDETrqF] 1	
			Dl, ^Dsup [TBDcX] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25	Dl, ^Dsup [TBDcX] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25	Dl, ^Dsup [TBDcX] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25	
AF-BEGIN-DIALOGUE (rejected(user), dataRI) cnf			^Dl, ^Dsup [TBDcX] [SDETrqF] 1	^Dl, ^Dsup [TBDcX] [SDETrqF] 1	^Dl, ^Dsup [TBDcX] [SDETrqF] 1	
			Du, Dl, ^Dsup [TBDcX] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25	Du, Dl, ^Dsup [TBDcX] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25	Du, Dl, ^Dsup [TBDcX] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25	
AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) cnf			Dl, ^Dsup [TBDcX] [CRBrs] [SDETrqF] [ABDET] [DELBRANCH] [TREERESET] 25	Dl, ^Dsup [TBDcX] [CRBrs] [SDETrqF] [ABDET] [DELBRANCH] [TREERESET] 25	Dl, ^Dsup [TBDcX] [CRBrs] [SDETrqF] [ABDET] [DELBRANCH] [TREERESET] 25	

Table A.13/X.862 (sheet 30 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
	Dsup ^Nrn, ^Da [ABDrsRUd] [SDETrqF] 1				Dsup ^Nrn, ^Da [ABDrsRUr] [SDETrqRBC] [REJTRAN] [TREERESET] 1		
^Dsup [VDaT] 11		^Dsup Dcr [TBDcX] [VDcrF] [VDaT] 15 ^Dsup ^Dcr [VDaT] 15				^Dsup [VDaT] 20.1	
^Dsup [TBDcX] [SDETrqF] 1		^Dsup [TBDcX] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25				^Dsup [TBDcXr] [SDETrqRB] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	
^Dsup [TBDcX] [SDETrqF] 1		^Dsup Du [TBDcX] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25				^Dsup Du [TBDcXr] [SDETrqRB] [ABDET] [INITRB] [OWEDONE] [COUNTRB] 23.2	
		^Dsup [TBDcX] [CRBrs] [SDETrqF] [ABDET] [DELBRANCH] [TREERESET] 25				^Dsup [TBDcXr] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	

Table A.13/X.862 (sheet 31 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
SAF-ASSOCIATION-LOST ind			^Dl [TBDcRP] 1	^Dl [TBDcRP] 1	^Dl [TBDcRP] 1	
			Dl [TBDcRP] [ABDET] [DELBRANCH] [TREERESET] 25	Dl [TBDcRP] [ABDET] [DELBRANCH] [TREERESET] 25	Dl [TBDcRP] [ABDET] [DELBRANCH] [TREERESET] 25	
TP-END-DIALOGUE (confirmation = FALSE) req			^Du, ^Dl, Dsup ^Ncr [DELIMIT] [AEDrqF] [SDETrqF] 1		Dsh, ^Du ^Dl, Dsup [AEDrqF] [SDETrqF] 1	
			Du, ^Dl, Dsup ^Ncr [DELIMIT] [AEDrqF] [SDETrqBF] 1		Dsh, Du ^Dl, Dsup [AEDrqF] [SDETrqBF] 1	
			^Dl, ^Dsup ^Dcr [AEDrqF] [SDETrqF] 1		Dsh, ^Dl, ^Dsup ^Dcr [AEDrqF] [SDETrqF] 1	
TP-END-DIALOGUE (confirmation = TRUE) req			^Dl, Dsup ^Ncr [DELIMIT] [AEDrq] 11		Dsh, ^Dl, Dsup ^Ncr [AEDrq] 11	
			^Dl, ^Dsup ^Dcr [AEDrq] 11		Dsh, ^Dl, ^Dsup ^Dcr [AEDrq] 11	
AF-END-DIALOGUE (confirmation = FALSE) ind			Dsh, ^Dl, Dsup ^Ncr [TEDi] [SDETrqF] 1	^Dl, Dsup ^Ncr [TEDi] [SDETrqF] 1	^Dl, Dsup ^Ncr [TEDi] [SDETrqF] 1	
			Dsh, ^Dl, ^Dsup ^Dcr [TEDi] [SDETrqF] 1	^Dl, ^Dsup ^Dcr [TEDi] [SDETrqF] 1	^Dl, ^Dsup ^Dcr [TEDi] [SDETrqF] 1	
			Dsh, Dx [TPABiBTED] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25		Dsh, Dx [TPABiBTED] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25	

Table A.13/X.862 (sheet 32 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
[TBDcRP] 1		[TBDcRP] [ABDET] [DELBRANCH] [TREERESET] 25				[TBDcRPr] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	
Dsh [TEDi] [SDETrqF] 1		Dsup Dsh, Dx [TPABiBTED] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25				Dsup Dsh, Dx [TPABiBTEDr] [SDETrqRB] [ABDET] [INITRB] [OWEDONE] [COUNTRB] 23.2	

Table A.13/X.862 (sheet 33 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
AF-END-DIALOGUE (confirmation = TRUE) ind			Dsh, ^Dl, Dsup ^Ncr [TEDi] 12	^Dl, Dsup ^Ncr [TEDi] 12	Dsh, ^Dl Denb=1 [DECDENB] 2	
			Dsh, ^Dl, ^Dsup ^Dcr [TEDi] 12	^Dl, ^Dsup ^Dcr [TEDi] 12	Dsh, ^Dl Denb>1 [DECDENB] 4	
			Dsh, Dx [TPABiBTED] [SDETrqRBR] [ABDET] [DELBRANCH] [TREERESET] 25		Dsh, Dx Denbb=0 [TPABiBTED] [SDETrqRBR] [ABDET] [DELBRANCH] [TREERESET] 25	
					Dsh, Dx Denbb>0 [DECDENB] 4	
					^Dsh, ^Dl [TEDi] [VDecT] 2	
TP-END-DIALOGUE rsp						
AF-END-DIALOGUE cnf						
TP-U-ERROR req			Dsh, Dsup ^Ncr [DELIMIT] [AUErq] [VDenbINC] 4	Dsup ^Ncr [DELIMIT] [AUErq] 4	Dsh, Dsup [AUErq] [VDenbINC] 4	
			Dsh, ^Dsup [AUErq] [VDenbINC] 4		Dsh, ^Dsup [AUErq] [VDenbINC] 4	
			^Dsh, Dsup ^Ncr [DELIMIT] [AUErq] 2			
			^Dsh, ^Dsup [AUErq] 2	^Dsup [AUErq] 4		

Table A.13/X.862 (sheet 34 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
Dsh Denb=0 [TPABiED] [SDETrqF] 1 Dsh Denb>0 [DECDENB] 11		Dsh, Dx ^Dsup Denbb=0 [TPABiBTED] [SDETrqRBR] [ABDET] [DELBRANCH] [TREERESET] 25 Du, Dsh ^Dcr, ^Dbcr Denbb>0 [DECDENB] 15				Dsh, Dx ^Dsup Denbb=0 [TPABiBTEDr] [SDETrqRBR [ABDET] [INITRB] [OWEDONE] [COUNTRB] 23.2 Du, Dsh ^Dbcr Denbb>0 [DECDENB] 20.1	
	[DELIMIT] [AEDrs] [SDETrqF] 1						
[TEDc] [SDETrqF] 1							
	Dsh [DELIMIT] [AUErq] 2 ^Dsh [DELIMIT] [AUErq] [VDecT] 2						

Table A.13/X.862 (sheet 35 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
AF-U-ERROR ind			Dsh, Dsup ^Da [TUEi] [VDepnbINC] 2	Dsup [TUEi] 3	Dsh, Dsup [TUEi] [AUErs] 4	
			Dsh, Dsup Da [TUEi] [AUErs] 2			
			Dsh, ^Dsup ^Dcr [TUEi] [AUErs] 2		Dsh, ^Dsup ^Dcr [TUEi] [AUErs] 4	
			^Dsh, Dsup [TUEi] 5		^Dsh, Dsup 4	
			^Dsh, ^Dsup ^Dcr [TUEi] 5	^Dsup ^Dcr [TUEi] 3	^Dsh, ^Dsup ^Dcr 4	
AF-U-ERROR cnf					Dsh Denb=1 [DECDENB] 2	
					Dsh Denb>1 [DECDENB] 4	
TP-U-ABORT req			^Du, ^Dl, Dsup ^Ncr [DELIMIT] [AABrqUd] [SDETrqF] 1	^Du, ^Dl, Dsup ^Ncr [DELIMIT] [AABrqUd] [SDETrqF] 1	^Du, ^Dl, Dsup [AABrqUd] [SDETrqF] 1	^Du, ^Dl, Dsup ^Ncr [DELIMIT] [AABrqUd] [SDETrqF] 1
			Du, ^Dl, Dsup ^Ncr [DELIMIT] [AABrqUd] [SDETrqBF] 1	Du, ^Dl, Dsup ^Ncr [DELIMIT] [AABrqUd] [SDETrqBF] 1	Du, ^Dl, Dsup [AABrqUd] [SDETrqBF] 1	Du, ^Dl, Dsup ^Ncr [DELIMIT] [AABrqUd] [SDETrqBF] 1
			^Dl, ^Dsup [AABrqUd] [SDETrqF] 1	^Dl, ^Dsup [AABrqUd] [SDETrqF] 1	^Dl, ^Dsup [AABrqUd] [SDETrqF] 1	^Dl, ^Dsup [AABrqUd] [SDETrqF] 1
			Dl, Dsup ^Ncr [DELIMIT] [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 2	Dl, Dsup ^Ncr [DELIMIT] [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 3	Dl, Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 4	Dl, Dsup ^Ncr [DELIMIT] [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 5
			Dl, ^Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 2	Dl, ^Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 3	Dl, ^Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 4	Dl, ^Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 5

(Continued on sheet 37 of 98)

Table A.13/X.862 (sheet 36 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP- COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
Dsh Denb=0 [TUEi] 2		^Dcr [SETDIAGUC] [TRBi] [INITRB] [OWEDONE] 15				[SETDIAGUC] [TRBi] [INITRB] [OWEDONE] 20.1	
Dsh Denb>0 [TUEi] 4							
^Dsh [TUEi] [VDecF] 3							
Dsh Denb>0 [DECDENB] 11		Dsh Denb>0 [DECDENB] 15				Dsh Denb>0 [DECDENB] 20.1	
^Du, Dsup [AABrqUd] [SDETrqF] 1	^Du, Dsup [DELIMIT] [AABrqUd] [SDETrqF] 1	Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 15	Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 16.1	Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 17	Dsup ^Ncr [DELIMIT] [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 18	Dsup, Nfa ^Ncr [DELIMIT] [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 20.1	Dsup, Nfa ^Ncr [DELIMIT] [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 20.2
Du, Dsup [AABrqUd] [SDETrqBF] 1	Du, Dsup [DELIMIT] [AABrqUd] [SDETrqBF] 1						
^Dsup [AABrqUd] [SDETrqF] 1	^Dsup [AABrqUd] [SDETrqF] 1	^Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 15	^Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 16.1	^Dsup ^Droi, ^Deei [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 17	^Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 18	^Dsup, Nfa [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 20.1	^Dsup, Nfa ^Droi, ^Deei [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 20.2

Table A.13/X.862 (sheet 37 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates Event		Dsup		^Dsh		^Dsh
TP-U-ABORT req *(Concluded 2 of 2)*						
AF-ABORT (user, dataRI) ind			^Dl, Dsup [TUABi] [SDETrqF] 1	^Dl, Dsup [TUABi] [SDETrqF] 1	^Dl, Dsup [TUABi] [SDETrqF] 1	^Dl [TUABi] [SDETrqF] 1
			^Dl, ^Dsup ^Dcr [TUABi] [SDETrqF] 1	^Dl, ^Dsup ^Dcr [TUABi] [SDETrqF] 1	^Dl, ^Dsup ^Dcr [TUABi] [SDETrqF] 1	
			Dl, ^Dsup ^Dcr, ^Dbcr [TUABi] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25	Dl, ^Dsup ^Dcr, ^Dbcr [TUABi] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25	Dl, ^Dsup ^Dcr, ^Dbcr [TUABi] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25	Dl, ^Dsup ^Dbcr [TUABi] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25
AF-ABORT (provider, abortRI) ind or A-ABORT ind or A-ABORT req or A-P-ABORT ind or A-RELEASE (result = affirmative) rsp or A-RELEASE (result = affirmative) cnf		1	^Dl [SETDIAGTP] [TPABi] 1	^Dl [SETDIAGTP] [TPABi] 1	^Dl [SETDIAGTP] [TPABi] 1	^Dl [SETDIAGTP] [TPABi] 1
			Dl, Dsup Ncr [SETDIAGTP] [TPABi] [REJTRAN] [TREERESET] 1	Dl, Dsup Ncr [SETDIAGTP] [TPABi] [REJTRAN] [TREERESET] 1		Dl, Dsup Ncr [SETDIAGTP] [TPABi] [REJTRAN] [TREERESET] 1
			Dl, Dsup ^Ncr [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dl, Dsup ^Ncr [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dl, Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dl, Dsup ^Ncr [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8
(Continued on sheet 39 of 98)			Dl, ^Dsup ^Dimpl [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup ^Dimpl [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup ^Dimpl [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup ^Dimpl [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2

Table A.13/X.862 (sheet 38 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
				^Dsup, Droi Nfa [ABTPSUI] [NOTCHAIN] 17			^Dsup, Droi Nfa [ABTPSUI] [NOTCHAIN] 20.2
				^Dsup, Deei Nfa [ABTPSUI] [NOTCHAIN] 17			^Dsup, Deei Nfa [ABTPSUI] [NOTCHAIN] 20.2
[TUABi] [SDETrqF] 1	[TUABi] [SDETrqF] 1	^Dsup ^Dcr, ^Dbcr [TUABi] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25				^Dsup ^Dbcr [TUABiR] [SDETrqRB] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	
[SETDIAGTP] [TPABi] 1	[SETDIAGTP] [TPABi] 1	Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8
		^Dsup [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, ^Deei, ^Droi [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, ^Deei, ^Droi [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2

Table A.13/X.862 (sheet 39 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates Event		Dsup		^Dsh		^Dsh
AF-ABORT (provider, abortRI) ind or A-ABORT ind or A-ABORT req or A-P-ABORT ind or A-RELEASE (result = affirmative) rsp or A-RELEASE (result = affirmative) cnf *(Concluded 2 of 2)*			Dl, ^Dsup Dimpl [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup Dimpl [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup Dimpl [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup Dimpl [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2
Protocol error or Internal error		[SETDIAG] [AABrqPa] 1	^Dl [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] 1	^Dl [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] 1	^Dl [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] 1	^Dl [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] 1
			Dl, Dsup Ncr [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [REJTRAN] [TREERESET] 1	Dl, Dsup Ncr [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [REJTRAN] [TREERESET] 1		Dl, Dsup Ncr [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [REJTRAN] [TREERESET] 1
			Dl, Dsup ^Ncr [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dl, Dsup ^Ncr [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dl, Dsup [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dl, Dsup ^Ncr [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8
(Continued on sheet 41 of 98)			Dl, ^Dsup ^Dimpl [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup ^Dimpl [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup ^Dimpl [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup ^Dimpl [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2

Table A.13/X.862 (sheet 40 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP- COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
				^Dsup, Droi [SETDIAGTP] [TPABi] [ABDET] [NOTCHAIN] 17			^Dsup, Droi [SETDIAGTP] [TPABi] [ABDET] [NOTCHAIN] [VNfaT] 20.2
				^Dsup, Deei [SETDIAGTP] [TPABi] [ABDET] [NOTCHAIN] 17			^Dsup, Deei [SETDIAGTP] [TPABi] [ABDET] [NOTCHAIN] [VNfaT] 20.2
[SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] 1	[SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] 1	Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [SETDIAG] [AABrqPa] [INITRB] [OWEDONE] 23.8	Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [SETDIAG] [AABrqPa] [INITRB] [OWEDONE] 23.8	Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [SETDIAG] [AABrqPa] [INITRB] [OWEDONE] 23.8	Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [SETDIAG] [AABrqPa] [INITRB] [OWEDONE] 23.8	Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [SETDIAG] [AABrqPa] [INITRB] [OWEDONE] 23.8	Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [SETDIAG] [AABrqPa] [INITRB] [OWEDONE] 23.8
		^Dsup [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [ABDET] [NOTCHAIN] [SETDIAG] [AABrqPa] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, ^Droi, ^Deei [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [ABDET] [NOTCHAIN] [SETDIAG] [AABrqPa] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [ABDET] [NOTCHAIN] [SETDIAG] [AABrqPa] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, ^Droi, ^Deei [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [ABDET] [NOTCHAIN] [SETDIAG] [AABrqPa] [INITRB] [OWEDONE] [COUNTRB] 23.2

Table A.13/X.862 (sheet 41 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
Protocol error or Internal error *(Concluded 2 of 2)*			Dl, ^Dsup Dimpl [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup Dimpl [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup Dimpl [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup Dimpl [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2
TP-GRANT-CONTROL req			^Dsh, Dsup ^Ncr [DELIMIT] [AGCrq] [VDecF] 3			Dsup ^Ncr [DELIMIT] [AGCrq] [VDecF] 3
			^Dsh, ^Dsup [AGCrq] [VDecF] 3			^Dsup [AGCrq] [VDecF] 3
AF-GRANT-CONTROL ind				Dsup [TGCi] [VDecT] 2	^Dsh, Dsup [TGCi] [VDecT] 2	
				^Dsup ^Dcr [TGCi] [VDecT] 2	^Dsh, ^Dsup ^Dcr [TGCi] [VDecT] 2	
TP-REQUEST-CONTROL req				Dsup ^Ncr [DELIMIT] [ARCrq] 3		
				^Dsup [ARCrq] 3		
AF-REQUEST-CONTROL ind			^Dsh, Dsup [TRCi] 2	3	^Dsh 4	
			^Dsh, ^Dsup ^Dcr [TRCi] 2			
TP-HANDSHAKE req			Dh, Dsup ^Ncr [DELIMIT] [AHSrq] 6		Dh, Dsh, Dsup [AHSrq] 6	
			Dh, ^Dsup [AHSrq] 6		Dh, Dsh, ^Dsup [AHSrq] 6	

Table A.13/X.862 (sheet 42 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
				^Dsup, Droi [SETDIAGTP] [TPABi] [ABDET] [NOTCHAIN] [VNfaT] 17			^Dsup, Droi [SETDIAGTP] [TPABi] [ABDET] [NOTCHAIN] [VNfaT] 20.2
				^Dsup, Deei [SETDIAGTP] [TPABi] [ABDET] [NOTCHAIN] [VNfaT] 17			^Dsup, Deei [SETDIAGTP] [TPABi] [ABDET] [NOTCHAIN] [VNfaT] 20.2
^Dsh 11		^Dsh ^Dcr 15				^Dsh 20.1	

Table A.13/X.862 (sheet 43 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
AF-HANDSHAKE ind			Dh, Dsh, Dsup [THSi] 7	Dh, Dsup [THSi] 7	Dh, Dsh, Dsup Denb=1 [DECDENB] 2	
					Dh, Dsh, Dsup Denb>1 [DECDENB] 4	
					Dh, ^Dsh, Dsup [THSi] [VDecT] 2	
			Dh, Dsh, ^Dsup ^Dcr [THSi] 7	Dh, ^Dsup ^Dcr [THSi] 7	Dh, Dsh, ^Dsup ^Dcr Denb=1 [DECDENB] 2	
					Dh, Dsh, ^Dsup ^Dcr Denb>1 [DECDENB] 4	
					Dh, ^Dsh, ^Dsup ^Dcr [THSi] [VDecT] 2	
TP-HANDSHAKE-AND-GRANT-CONTROL req			Dh, ^Dsh, Dsup ^Ncr [DELIMIT] [AHSGCrq] [VDecF] 13			
			Dh, ^Dsh, ^Dsup [AHSGCrq] [VDecF] 13			
AF-HANDSHAKE-AND-GRANT- CONTROL ind				Dh, Dsup [THSGCi] 14	Dh, ^Dsh, Dsup [THSGCi] [VDecT] 2	
				Dh, ^Dsup ^Dcr [THSGCi] 14	Dh, ^Dsh, ^Dsup ^Dcr [THSGCi] [VDecT] 2	

Table A.13/X.862 (sheet 44 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
Dh, Dsh Denb=0 [THSi] 10		Dsh ^Dcr [SETDIAGUC] [TRBi] [INITRB] [OWEDONE] 15				Dsh [SETDIAGUC] [TRBi] [INITRB] [OWEDONE] 20.1	
Dh, Dsh Denb>0 [DECDENB] 11							

Table A.13/X.862 (sheet 45 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-BEGIN-TRANSACTION (check-ready-directions = true or check-ready-directions parameter is absent) req			Du, ^Dl, ^Dsup ^Nt, ^Da Dgrp=1 ^Ntch Nr [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup Dsh, ^Nt, ^Da Dgrp=1 ^Ntch Nr [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dgrp=1 ^Ntch Nr [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dsh, Dgrp=1 ^Ntch Nr [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, ^Da Dgrp=1 ^Nr, ^Ni, ^Nlf [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, ^Da Dsh, Dgrp=1 ^Nr, ^Ni, ^Nlf [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dgrp=1 ^Nr, ^Ni, ^Nlf [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dsh, Dgrp=1 ^Nr, ^Ni, ^Nlf [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, ^Da Dgrp=2 ^Nex, ^N2exp Nr [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, ^Da Dsh, Dgrp=2 ^Nex, ^N2exp Nr [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dgrp=2 ^Nex, ^N2exp Nr [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dsh, Dgrp=2 ^Nex, ^N2exp Nr [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	

(Continued on sheet 47 of 98)

Table A.13/X.862 (sheet 46 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 47 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-BEGIN-TRANSACTION (check-ready-directions = true or check-ready-directions parameter is absent) req *(Continued 2 of 5)*			Du, ^Dl, ^Dsup ^Nt, ^Da Dgrp=2 ^Nr, ^Ni, ^Nlf [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, ^Da Dsh, Dgrp=2 ^Nr, ^Ni, ^Nlf [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dgrp=2 ^Nr, ^Ni, ^Nlf [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dsh, Dgrp=2 ^Nr, ^Ni, ^Nlf [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, ^Da Dgrp=3 ^Ntch [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, ^Da Dsh, Dgrp=3 ^Ntch [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dgrp=3 ^Ntch [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dsh, Dgrp=3 ^Ntch [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, ^Da Dgrp=3 Ntch ^Nex [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, ^Da Dsh, Dgrp=3 Ntch ^Nex [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
(Continued on sheet 49 of 98)			Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dgrp=3 Ntch ^Nex [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dsh, Dgrp=3 Ntch ^Nex [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	

Table A.13/X.862 (sheet 48 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 49 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-BEGIN-TRANSACTION (check-ready-directions = true or check-ready-directions parameter is absent) req *(Continued 3 of 5)*			Du, ^Dl, ^Dsup ^Nt, ^Da Dgrp=4 ^Nex [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, ^Da Dsh, Dgrp=4 ^Nex [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dgrp=4 ^Nex [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dsh, Dgrp=4 ^Nex [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, ^Da Dgrp=5 ^Ntch [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, ^Da Dsh, Dgrp=5 ^Ntch [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dgrp=5 ^Ntch [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dsh, Dgrp=5 ^Ntch [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, ^Da Dgrp=5 Ntch ^Nsopex [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, ^Da Dsh, Dgrp=5 Ntch ^Nsopex [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
(Continued on sheet 51 of 98)			Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dgrp=5 Ntch ^Nsopex [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dsh, Dgrp=5 Ntch ^Nsopex [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	

Table A.13/X.862 (sheet 50 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 51 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-BEGIN-TRANSACTION (check-ready-directions = true or check-ready-directions parameter is absent) req *(Continued 4 of 5)*			Du, ^Dl, ^Dsup ^Nt, ^Da Dgrp=7 ^Ntch [VDgrp6] [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, ^Da Dsh, Dgrp=7 ^Ntch [VDgrp6] [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dgrp=7 ^Ntch [VDgrp6] [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dsh, Dgrp=7 ^Ntch [VDgrp6] [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, ^Da Dgrp=7 Ntch ^Nsopex [VDgrp6] [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, ^Da Dsh, Dgrp=7 Ntch ^Nsopex [VDgrp6] [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dgrp=7 Ntch ^Nsopex [VDgrp6] [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dsh, Dgrp=7 Ntch ^Nsopex [VDgrp6] [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, ^Da Dgrp=9 ^Nsopex [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, ^Da Dsh, Dgrp=9 ^Nsopex [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
(Continued on sheet 53 of 98)			Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dgrp=9 ^Nsopex [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dsh, Dgrp=9 ^Nsopex [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	

Table A.13/X.862 (sheet 52 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 53 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates Event		Dsup		^Dsh		^Dsh
TP-BEGIN-TRANSACTION (check-ready-directions = true or check-ready-directions parameter is absent) req *(Concluded 5 of 5)*			Du, ^Dl, ^Dsup ^Nt, ^Da Dgrp=11 ^Nsopex [VDgrp10] [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, ^Da Dsh, Dgrp=11 ^Nsopex [VDgrp10] [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
			Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dgrp=11 ^Nsopex [VDgrp10] [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2		Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dsh, Dgrp=11 ^Nsopex [VDgrp10] [TREESET] [ADDBRSB] [BEGTRANS] [CBErq] 2	
TP-BEGIN-TRANSACTION (check-ready-directions = false) req			Du, ^Dl, ^Dsup ^Nt, ^Da Dgrp=7 ^Ntch [VDgrp8] [TREESET] [ADDBRSB] [BEGTRANS] [ABTrq] 2		Du, ^Dl, ^Dsup ^Nt, ^Da Dsh, Dgrp=7 ^Ntch [VDgrp8] [TREESET] [ADDBRSB] [BEGTRANS] [ABTrq] 2	
			Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dgrp=7 ^Ntch [VDgrp8] [TREESET] [ADDBRSB] [BEGTRANS] [ABTrq] 2		Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dsh, Dgrp=7 ^Ntch [VDgrp8] [TREESET] [ADDBRSB] [BEGTRANS] [ABTrq] 2	
			Du, ^Dl, ^Dsup ^Nt, ^Da Dgrp=11 ^Ntch ^Nsopex [VDgrp12] [TREESET] [ADDBRSB] [BEGTRANS] [ABTrq] 2		Du, ^Dl, ^Dsup ^Nt, ^Da Dsh, Dgrp=11 ^Ntch ^Nsopex [VDgrp12] [TREESET] [ADDBRSB] [BEGTRANS] [ABTrq] 2	
			Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dgrp=11 ^Ntch ^Nsopex [VDgrp12] [TREESET] [ADDBRSB] [BEGTRANS] [ABTrq] 2		Du, ^Dl, ^Dsup ^Nt, Ptok, Da Dsh, Dgrp=11 ^Ntch ^Nsopex [VDgrp12] [TREESET] [ADDBRSB] [BEGTRANS] [ABTrq] 2	

Table A.13/X.862 (sheet 54 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl

Table A.13/X.862 (sheet 55 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
C-BEGIN ind		Dsh Dgrp=1 [TREESET] [TBDiSAVE] [ADDBRSP] [VNtpsuiT] 2	Du, Dsh ^Dl, Dsup Nr [TPABiBTR] [AABrqPrTR] [SDETrqRBC] 1	Du, ^Dl, Dsup Nr [TPABiBTR] [AABrqPrTR] [SDETrqRBC] 1	Du, ^Dl, Dsup Nr [TPABiBTR] [AABrqPrTR] [SDETrqRBC] 1	
		^Dsh Dgrp=1 [TREESET] [TBDiSAVE] [ADDBRSP] [VNtpsuiT] [VDecF] 3				
			Du, Dsh ^Dl, Dsup ^Nr Dgrp=2 [VDgrp3] [TREESET] [TBTi] [ADDBRSP] 2	Du, ^Dl, Dsup ^Nr Dgrp=2 [VDgrp3] [TREESET] [TBTi] [ADDBRSP] [VDecF] 3	Du, ^Dl, Dsup ^Nr Dgrp=2 [VDgrp3] [TREESET] [TBTi] [ADDBRSP] [VDecF] 4	
		Dsh Dgrp=3 [TREESET] [TBDiSAVE] [ADDBRSP] [VNtpsuiT] 2	Dsh Dgrp=3 [TREESET] [TBDiSAVE] [ADDBRSP] [VNtpsuiT] 3			
		^Dsh Dgrp=3 [TREESET] [TBDiSAVE] [ADDBRSP] [VNtpsuiT] [VDecF] 3		Du, ^Dl, Dsup ^Nr Dgrp=3 [TREESET] [TBDiSAVE] [ADDBRSP] [VNtpsuiT] [VDecF] 3	Du, ^Dl, Dsup ^Nr Dgrp=3 [TREESET] [TBDiSAVE] [ADDBRSP] [VNtpsuiT] [VDecF] 4	
(Continued on sheet 57 of 98)		Dsh Dgrp=4 [TREESET] [TBDiSAVE] [ADDBRSP] [VNtpsuiT] 2	Du, Dsh ^Dl, Dsup ^Nr Dgrp=4 [TREESET] [TBTi] [ADDBRSP] 2		Du, ^Dl, Dsup ^Nr Dgrp=4 [TREESET] [TBTi] [ADDBRSP] [VDecF] 4	

Table A.13/X.862 (sheet 56 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
Du, Dsh, Dsup [TPABiBTED] [CRBrq] [SDETrqRBC] 1							

Table A.13/X.862 (sheet 57 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
C-BEGIN ind *(Concluded 2 of 2)*		^Dsh Dgrp=4 [TREESET] [TBDiSAVE] [ADDBRSP] [VNtpsuiT] [VDecF] 3		Du, ^Dl, Dsup ^Nr Dgrp=4 [TREESET] [TBTi] [ADDBRSP] [VDecF] 3		
AF-BEGIN-TRANSACTION ind			Du, Dsh ^Dl, Dsup Nr [TPABiBTR] [AABrqPrTR] [SDETrqRBC] 1	Du, ^Dl, Dsup Nr [TPABiBTR] [AABrqPrTR] [SDETrqRBC] 1	Du, ^Dl, Dsup Nr [TPABiBTR] [AABrqPrTR] [SDETrqRBC] 1	
			Du, Dsh ^Dl, Dsup ^Nr Dgrp=2 [VDgrp1] [TREESET] [TBTi] [ADDBRSP] 2	Du, ^Dl, Dsup ^Nr Dgrp=2 [VDgrp1] [TREESET] [TBTi] [ADDBRSP] 3	Du, ^Dl, Dsup ^Nr Dgrp=2 [VDgrp1] [TREESET] [TBTi] [ADDBRSP] 4	
C-BEGIN cnf			Dl, ^Dsup [VDbcrT] [VDxF] 2	Dl, ^Dsup [VDbcrT] [VDxF] 3	Dl, ^Dsup [VDbcrT] [VDxF] 4	Dl, ^Dsup [VDbcrT] [VDxF] 5
TP-DATA req			Dsup ^Ncr [DELIMIT] [UASErq] 2		Dsh, Dsup [UASErq] 4	
			^Dsup [UASErq] 2		Dsh, ^Dsup [UASErq] 4	
U-ASE ind			Dsh, Dsup [TDTi] 2	Dsup [TDTi] 3	4	
			Dsh, ^Dsup ^Dcr [TDTi] 2	^Dsup ^Dcr [TDTi] 3		

Table A.13/X.862 (sheet 58 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
Du, Dsh, Dsup [TPABiBTED] [CRBrq] [SDETrqRBC] 1							
		^Dsup [VDbcrT] [VDxF] 15				^Dsup [VDbcrT] [VDxF] 20.1	
					Dsup, Dsh ^Ncr [DELIMIT] [UASErq] 18		
					Dsup, ^Dsh ^Ncr, Ddp [DELIMIT] [UASErq] 18		
					^Dsup, Dsh [UASErq] 18		
Dsh Denb=0 [TDTi] 11		^Dsup Dsh ^Dcr [TDTi] 15				Dsh [SETDIAGUC] [TRBi] [INITRB] [OWEDONE] 20.1	
Dsh Denb>0 11		^Dsup ^Dsh ^Dcr, Ddp [TDTi] 15				^Dsh, Ddp [SETDIAGUC] [TRBi] [INITRB] [OWEDONE] 20.1	

Table A.13/X.862 (sheet 59 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-DEFERRED-END-DIALOGUE req			Dl, ^Dsup ^Dimpl ^De, ^Lddef [ADErq] [VDdefF] [VDeT] 2		Dsh, Dl, ^Dsup ^Dimpl ^De, ^Lddef [ADErq] [VDdefF] [VDeT] 4	
			Dl, ^Dsup ^Dimpl ^De, Lddef [VDdefT] [VDeT] 2		Dsh, Dl, ^Dsup ^Dimpl ^De, Lddef [VDdefT] [VDeT] 4	
			Dl, ^Dsup Dimpl ^De [ADErq] [VDdefF] [VDeT] 2		Dsh, Dl, ^Dsup Dimpl ^De [ADErq] [VDdefF] [VDeT] 4	
TP-DEFERRED-GRANT-CONTROL req	Q		^Dsh, Dl, ^Dsup ^Dimpl ^Dg, ^De ^Lddef [ADGrq] [VDgT] 2			
			^Dsh, Dl, ^Dsup ^Dimpl ^Dg, ^De Lddef [VDdefT] [VDgT] 2			
			^Dsh, Dl, ^Dsup Dimpl ^Dg, ^De [ADGrq] [VDgT] 2			
AF-DEFER (end-dialogue) ind			Dsh, Dl, Dsup ^De [TDEi] [VDeT] 2	Dl, Dsup ^De [TDEi] [VDeT] 3	Dl, Dsup ^De [TDEi] [VDeT] 4	
AF-DEFER (grant-control) ind				Dl, Dsup ^Dg, ^De [TDGi] [VDgT] 3	^Dsh, Dl, Dsup ^Dg, ^De [TDGi] [VDgT] 4	

Table A.13/X.862 (sheet 60 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
						Dsup Dsh ^De [SETDIAGUC] [TRBi] [INITRB] [OWEDONE] 20.1	

Table A.13/X.862 (sheet 61 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-PREPARE req			Dsh, Dl, Nr ^Ddef [APRrq] [VDpsT] 15		Dsh, Dl, Nr ^Ddef [APRrq] [VDpsT] 15	
			Dsh, Dl, Nr Ddef [DEFREQ] [APRrq] [VDpsT] 15		Dsh, Dl, Nr Ddef [DEFREQ] [APRrq] [VDpsT] 15	
			Dsh, Dl, ^Dsup Ni, Np ^Ddef [APRrq] [VDpsT] 15		Dsh, Dl, ^Dsup Ni, Np ^Ddef [APRrq] [VDpsT] 15	
			Dsh, Dl, ^Dsup Ni, Np Ddef [DEFREQ] [APRrq] [VDpsT] 15		Dsh, Dl, ^Dsup Ni, Np Ddef [DEFREQ] [APRrq] [VDpsT] 15	
			Dsh, Dl, Dsup Ddyn Nimpl [DELIMIT] [APRrq] [VDpsT] 15		Dsh, Dl, Dsup Ddyn Nimpl [DELIMIT] [APRrq] [VDpsT] 15	
TP-PREPARE (data-permitted = FALSE) req			^Dsh, Dl, Nr ^Ddef [APRrq] [VDpsT] 15			
			^Dsh, Dl, Nr Ddef [DEFREQ] [APRrq] [VDpsT] 15			
			^Dsh, Dl, ^Dsup Ni, Np ^Ddef [APRrq] [VDpsT] 15			
			^Dsh, Dl, ^Dsup Ni, Np Ddef [DEFREQ] [APRrq] [VDpsT] 15			
			^Dsh, Dl, Dsup Ddyn Nimpl [DELIMIT] [APRrq] [VDpsT] 15			

Table A.13/X.862 (sheet 62 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
					Dsh, Nr Ddyn [APRrq] [VDpsT] 16.1		
					Dsh, ^Dsup Ni, Np Ddyn [APRrq] [VDpsT] 16.1		
					Dsh, Dsup Ddyn [APRrq] [VDpsT] 16.1		
					^Dsh, Nr Ddyn, Dimpl Ddp [APRrq] [VDpsT] 16.1		
					^Dsh, ^Dsup Ni, Np Ddyn, Dimpl Ddp [APRrq] [VDpsT] 16.1		
					^Dsh, Dsup Ddyn, Ddp [APRrq] [VDpsT] 16.1		

Table A.13/X.862 (sheet 63 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates Event		Dsup		^Dsh		^Dsh
TP-PREPARE (data-permitted = TRUE) req			^Dsh, Dl, Nr ^Ddef [APRrq] [VDdpT] [VDpsT] 15			
			^Dsh, Dl, Nr Ddef [DEFREQ] [APRrq] [VDdpT] [VDpsT] 15			
			^Dsh, Dl, ^Dsup Ni, Np ^Ddef [APRrq] [VDdpT] [VDpsT] 15			
			^Dsh, Dl, ^Dsup Ni, Np Ddef [DEFREQ] [APRrq] [VDdpT] [VDpsT] 15			
			^Dsh, Dl, Dsup ^Ncr, Ddyn Nimpl [DELIMIT] [APRrq] [VDpsT] 15			
TP-COMMIT req			Dl, Nr ^Dcr, ^Ddef [VNcmtrT] [COUNTGE] [VNtT] 20.1		Dsh, Dl, Nr ^Dcr, ^Ddef [VNcmtrT] [COUNTGE] [VNtT] 20.1	
			Dl, Nr ^Dcr, Ddef [DEFREQ] [VNcmtrT] [COUNTGE] [VNtT] 20.1		Dsh, Dl, Nr ^Dcr, Ddef [DEFREQ] [VNcmtrT] [COUNTGE] [VNtT] 20.1	
			Dl, ^Dsup Ni, Np ^Dcr, ^Ddef [VNcmtrT] [COUNTGE] [VNtT] 20.1		Dsh, Dl, ^Dsup Ni, Np ^Dcr, ^Ddef [VNcmtrT] [COUNTGE] [VNtT] 20.1	
(Continued on sheet 65 of 98)			Dl, ^Dsup Ni, Np ^Dcr, Ddef [DEFREQ] [VNcmtrT] [COUNTGE] [VNtT] 20.1		Dsh, Dl, ^Dsup Ni, Np ^Dcr, Ddef [DEFREQ] [VNcmtrT] [COUNTGE] [VNtT] 20.1	

Table A.13/X.862 (sheet 64 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
		^Dsup ^Dcr [VNcmtrT] [COUNTGE] [VNtT] 20.1	^Dsup ^Dcr [VNcmtrT] [COUNTGE] [VNtT] 20.1	^Dsup ^Dcr [VNcmtrT] [COUNTGE] [VNtT] 20.2	^Dsup, Ddyn ^Dcr [VNcmtrT] [COUNTGE] [VNtT] 20.1		

Table A.13/X.862 (sheet 65 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-COMMIT req *(Concluded 2 of 2)*			Dl, Dsup Nimpl ^Ncr [VNcmtrT] [COUNTGE] [VNtT] 20.1		Dsh, Dl, Dsup Nimpl ^Ncr [VNcmtrT] [COUNTGE] [VNtT] 20.1	
TP-ONE-PHASE req			Dl, Nr ^Dcr, ^Ddef [VNoprT] [COUNTGE] [VNtT] 20.1		Dsh, Dl, Nr ^Dcr, ^Ddef [VNoprT] [COUNTGE] [VNtT] 20.1	
			Dl, Nr ^Dcr, Ddef [DEFREQ] [VNoprT] [COUNTGE] [VNtT] 20.1		Dsh, Dl, Nr ^Dcr, Ddef [DEFREQ] [VNoprT] [COUNTGE] [VNtT] 20.1	
			Dl, ^Dsup Ni, Np ^Dcr, ^Ddef [VNoprT] [COUNTGE] [VNtT] 20.1		Dsh, Dl, ^Dsup Ni, Np ^Dcr, ^Ddef [VNoprT] [COUNTGE] [VNtT] 20.1	
			Dl, ^Dsup Ni, Np ^Dcr, Ddef [DEFREQ] [VNoprT] [COUNTGE] [VNtT] 20.1		Dsh, Dl, ^Dsup Ni, Np ^Dcr, Ddef [DEFREQ] [VNoprT] [COUNTGE] [VNtT] 20.1	
			Dl, Dsup Nimpl ^Ncr [VNoprT] [COUNTGE] [VNtT] 20.1		Dsh, Dl, Dsup Nimpl ^Ncr [VNoprT] [COUNTGE] [VNtT] 20.1	

Table A.13/X.862 (sheet 66 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
		Dsup, Ddyn ^Ncr [VNcmtrT] [COUNTGE] [VNtT] 20.1	Dsup ^Ncr [VNcmtrT] [COUNTGE] [VNtT] 20.1	Dsup ^Ncr [VNcmtrT] [COUNTGE] [VNtT] 20.2	Dsup ^Ncr [VNcmtrT] [COUNTGE] [VNtT] 20.1		
		^Dsup ^Dcr [VNoprT] [COUNTGE] [VNtT] 20.1	^Dsup ^Dcr [VNoprT] [COUNTGE] [VNtT] 20.1	^Dsup ^Dcr [VNoprT] [COUNTGE] [VNtT] 20.2	^Dsup, Ddyn ^Dcr [VNoprT] [COUNTGE] [VNtT] 20.1		
		Dsup, Ddyn ^Ncr [VNoprT] [COUNTGE] [VNtT] 20.1	Dsup ^Ncr [VNoprT] [COUNTGE] [VNtT] 20.1	Dsup ^Ncr [VNoprT] [COUNTGE] [VNtT] 20.2	Dsup ^Ncr [VNoprT] [COUNTGE] [VNtT] 20.1		

Table A.13/X.862 (sheet 67 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-READ-ONLY req			Dl, ^Dsup Ni, Np ^Dcr, ^Ddef [VNrorT] [COUNTGE] [VNtT] 20.1		Dsh, Dl, ^Dsup Ni, Np ^Dcr, ^Ddef [VNrorT] [COUNTGE] [VNtT] 20.1	
			Dl, ^Dsup Ni, Np ^Dcr, Ddef [DEFREQ] [VNrorT] [COUNTGE] [VNtT] 20.1		Dsh, Dl, ^Dsup Ni, Np ^Dcr, Ddef [DEFREQ] [VNrorT] [COUNTGE] [VNtT] 20.1	
			Dl, Dsup Drofu Nimpl ^Ncr [VNrorT] [COUNTGE] [VNtT] 20.1		Dsh, Dl, Dsup Drofu Nimpl ^Ncr [VNrorT] [COUNTGE] [VNtT] 20.1	
TP-EARLY-EXIT req			Dl, Dsup, Deefu ^Dcdfu,^Ncr ^Ldbd [VNeerT] [VNtT] [VDgF] [VDeF] [COUNTGE] 20.1		Dl, Dsup, Deefu ^Dcdfu ^Ldbd [VNeerT] [VNtT] [VDgF] [VDeF] [COUNTGE] 20.1	Dl, Dsup Deefu ^Dcdfu, ^Ncr ^Ldbd [VNeerT] [VNtT] [VDgF] [VDeF] [COUNTGE] 20.1
			Dl, Dsup, Deefu Dcdfu, ^Ncr ^Ldbd [VNeerT] [VNtT] [VDgF] [VDeF] [SAVECR] [COUNTGE] 20.1		Dl, Dsup, Deefu Dcdfu ^Ldbd [VNeerT] [VNtT] [VDgF] [VDeF] [SAVECR] [COUNTGE] 20.1	Dl, Dsup Deefu Dcdfu, ^Ncr ^Ldbd [VNeerT] [VNtT] [VDgF] [VDeF] [SAVECR] [COUNTGE] 20.1
			Dl, Dsup ^Ncr, ^Nfrb Ldbd, Deefu [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3	Dl, Dsup ^Ncr, ^Nfrb Ldbd, Deefu [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3	Dl, Dsup ^Nfrb Ldbd, Deefu [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3	Dl, Dsup ^Ncr, ^Nfrb Ldbd, Deefu [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3
			Dl, Dsup ^Ncr, Nfrb Ldbd, Deefu [CANCEL] 23.3	Dl, Dsup ^Ncr, Nfrb Ldbd, Deefu [CANCEL] 23.3	Dl, Dsup Nfrb Ldbd, Deefu [CANCEL] 23.3	Dl, Dsup ^Ncr, Nfrb Ldbd, Deefu [CANCEL] 23.3

(Continued on sheet 69 of 98)

Table A.13/X.862 (sheet 68 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP- COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
		^Dsup ^Dcr [VNrorT] [COUNTGE] [VNtT] 20.1	^Dsup ^Dcr [VNrorT] [COUNTGE] [VNtT] 20.1	^Dsup ^Dcr [VNrorT] [COUNTGE] [VNtT] 20.2	^Dsup, Ddyn ^Dcr [VNrorT] [COUNTGE] [VNtT] 20.1		
		Dsup, Ddyn Drofu ^Ncr [VNrorT] [COUNTGE] [VNtT] 20.1	Dsup Drofu ^Ncr [VNrorT] [COUNTGE] [VNtT] 20.1	Dsup Drofu ^Ncr [VNrorT] [COUNTGE] [VNtT] 20.2	Dsup Drofu ^Ncr [VNrorT] [COUNTGE] [VNtT] 20.1		
		Dsup, Deefu ^Dcdfu, ^Ncr ^Ldbd [VNeerT] [VNtT] [VDgF] [VDeF] [COUNTGE] 20.1	Dsup, Deefu ^Dcdfu, ^Ncr ^Ldbd [VNeerT] [VNtT] [VDgF] [VDeF] [COUNTGE] 20.1	Ni, ^Dsup, Deei [COUNTGE] 20.2	Dsup, Deefu ^Dcdfu, ^Ncr ^Ldbd [VNeerT] [VNtT] [VDgF] [VDeF] [COUNTGE] 20.1		
		Dsup, Deefu Dcdfu, ^Ncr ^Ldbd [VNeerT] [VNtT] [VDgF] [VDeF] [SAVECR] [COUNTGE] 20.1	Dsup, Deefu Dcdfu, ^Ncr ^Ldbd [VNeerT] [VNtT] [VDgF] [VDeF] [SAVECR] [COUNTGE] 20.1		Dsup, Deefu Dcdfu, ^Ncr ^Ldbd [VNeerT] [VNtT] [VDgF] [VDeF] [SAVECR] [COUNTGE] 20.1		
		Dsup ^Nfrb ^Ncr Ldbd, Deefu [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3	Dsup ^Nfrb ^Ncr Ldbd, Deefu [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3	Dsup ^Nfrb ^Ncr Ldbd, Deefu [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3	Dsup ^Nfrb ^Ncr Ldbd, Deefu [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3		
		Dsup Nfrb, ^Ncr Ldbd, Deefu [CANCEL] 23.3	Dsup Nfrb, ^Ncr Ldbd, Deefu [CANCEL] 23.3	Dsup Nfrb, ^Ncr Ldbd, Deefu [CANCEL] 23.3	Dsup Nfrb, ^Ncr Ldbd, Deefu [CANCEL] 23.3		

Table A.13/X.862 (sheet 69 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-EARLY-EXIT req *(Concluded 2 of 2)*						
AF-PREPARE ind			Dsh, Dl, Dsup [TPRi] [VNpT] 18		Dsh, Dl, Dsup [SETDIAGUC] [TRBi] [INITRB] [OWEDONE] 4	
			Dsh, Dl, ^Dsup Dimpl [TPRi] [VNpT] 18			
AF-PREPARE (data-permitted = FALSE) ind				Dl, Dsup Dimpl [TPRi] [VNpT] 18	^Dsh, Dl, Dsup [SETDIAGUC] [TRBi] [INITRB] [OWEDONE] 4	
				Dl, ^Dsup [TPRi] [VNpT] 18	^Dsh, Dl, ^Dsup [SETDIAGUC] [TRBi] [INITRB] [OWEDONE] 4	
AF-PREPARE (data-permitted = TRUE) ind				Dl, Dsup [TPRi] [VNpT] [VDdpT] 18	^Dsh, Dl, Dsup [SETDIAGUC] [TRBi] [INITRB] [OWEDONE] 4	

Table A.13/X.862 (sheet 70 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
				Ni, ^Dsup, ^Deei ^Nfrb [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1			
				Ni, ^Dsup, ^Deei Nfrb [RBREQ] 23.1			
				Ni, ^Dsup, Deei ^Nfrb [TRBi] [INITRB] [OWEDONE] [COUNTRB] 23.2			
				Ni, ^Dsup, Deei Nfrb [COUNTRB] 23.2			
		Dsup [TPRi] [VNpT] 16.1				Dsup, Nimpl [VNpT] 20.1	
		^Dsup, Dimpl ^Dcr [TPRi] 16.1				^Dsup 20.1	
		Dl, Dsup [TPRi] [VNpT] 16.1				^Dsup ^Dsh 20.1	
		Dl, ^Dsup [TPRi] [VNpT] 16.1				Dsup Nimpl 20.1	

Table A.13/X.862 (sheet 71 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
C-READY ind			Dl, Dsup Ddyn Drrec [TRYi] [VNpT] [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] [TREERESET] 17	Dl, Dsup Ddyn Drrec [TRYi] [VNpT] [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] [TREERESET] 17	Dl, Dsup Drrec [SETDIAGUC] [TRBi] [INITRB] [OWEDONE] 4	
			Dl, ^Dsup, ^Dcr Drrec Dimpl Dch [TRYi] [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] 17	Dl, ^Dsup, ^Dcr Drrec Dimpl Dch [TRYi] [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] 17		
			Dl, ^Dsup, ^Dcr Drrec Dimpl ^Dch [TRYi] [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] [TREERESET] 17	Dl, ^Dsup, ^Dcr Drrec Dimpl ^Dch [TRYi] [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] [TREERESET] 17		
C-NOCHANGE (result-requested) ind			Dl, ^Dsup, Dopfu Dimpl Drrec Dch [TOPi] [VNopiINC] [VDopiT] [COUNTRDY] 17	Dl, ^Dsup Dopfu, Dimpl Drrec Dch [TOPi] [VNopiINC] [VDopiT] [COUNTRDY] 17	Dl, Dsup Drrec [SETDIAGUC] [TRBi] [INITRB] [OWEDONE] 4	
			Dl, ^Dsup, Dopfu Dimpl Drrec ^Dch [TOPi] [VNopiINC] [VDopiT] [COUNTRDY] [TREERESET] 17	Dl, ^Dsup Dopfu, Dimpl Drrec ^Dch [TOPi] [VNopiINC] [VDopiT] [COUNTRDY] [TREERESET] 17		
			Dl, Dsup, Ddyn Drrec [TOPi] [VNopiINC] [VDopiT] [VNpT] [COUNTRDY] [TREERESET] 17	Dl, Dsup, Ddyn Drrec [TOPi] [VNopiINC] [VDopiT] [VNpT] [COUNTRDY] [TREERESET] 17		

(Continued on sheet 73 of 98)

Table A.13/X.862 (sheet 72 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP- COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
		Dsup Ddyn Drrec [TRYi] [VNpT] [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] [TREERESET] 17	Dsup Ddyn Drrec [TRYi] [VNpT] [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] [TREERESET] 17		Dsup Ddyn ^Dcr, Drrec [TRYi] [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] [TREERESET] 17	Dsup, Ddyn ^Dcr, Drrec [VNpT] [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] 20.2	
		^Dsup, ^Dcr Drrec Dch [TRYi] [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] 17	^Dsup, ^Dcr Drrec Dch [TRYi] [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] 17		^Dsup Drrec Dimpl Dch [TRYi] [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] 17	^Dsup Drrec [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] 20.2	
		^Dsup, ^Dcr Drrec ^Dch [TRYi] [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] [TREERESET] 17	^Dsup, ^Dcr Drrec ^Dch [TRYi] [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] [TREERESET] 17		^Dsup Drrec Dimpl ^Dch [TRYi] [CRDYSET] [VNrdyiINC] [VDrdyiT] [COUNTRDY] [TREERESET] 17		
					^Dsup, Dopfu Drrec Dch [TOPi] [VNopiINC] [VDopiT] [COUNTRDY] 17	^Dsup, Dopfu Drrec [VNopiINC] [VDopiT] [COUNTRDY] 20.2	
		^Dsup, Dopfu Drrec [TOPi] [VNopiINC] [VDopiT] [COUNTRDY] [TREERESET] 17	^Dsup, Dopfu Drrec [TOPi] [VNopiINC] [VDopiT] [COUNTRDY] [TREERESET] 17		^Dsup, Dopfu Drrec ^Dch [TOPi] [VNopiINC] [VDopiT] [COUNTRDY] [TREERESET] 17		
		Dsup, Ddyn Drrec [TOPi] [VNopiINC] [VDopiT] [VNpT] [COUNTRDY] [TREERESET] 17	Dsup, Ddyn Drrec [TOPi] [VNopiINC] [VDopiT] [COUNTRDY] [TREERESET] 17		Dsup, Ddyn Drrec [TOPi] [VNopiINC] [VDopiT] [COUNTRDY] [TREERESET] 17	Dsup, Ddyn Drrec [VNopiINC] [VDopiT] [COUNTRDY] 20.2	

Table A.13/X.862 (sheet 73 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
C-NOCHANGE (result-requested) ind *(Concluded 2 of 2)*			Dl, Dsup, ^Ddyn Drrec ^Do, ^Nch [TOPi] [VNopiINC] [VDopiT] [VNpT] [COUNTRDY] [TREERESET] 17	Dl, Dsup Drrec, ^Ddyn ^Do, ^Nch [TOPi] [VNopiINC] [VDopiT] [VNpT] [COUNTRDY] [TREERESET] 17		
AF-NOCHANGE (result-requested) ind			Dl, Dsup ^Do, Nch [TOPi] [VNopiINC] [VDopiT] [SAVEAAIDN] [VNpT] [COUNTRDY] 17	Dl, Dsup Drrec, ^Do, Nch [TOPi] [VNopiINC] [VDopiT] [SAVEAAIDN] [VNpT] [COUNTRDY] 17	Dl, Dsup Drrec, ^Do, Nch Drrec [SETDIAGUC] [TRBi] [INITRB] [OWEDONE] 4	
C-NOCHANGE (result-not-required) ind			Dl, ^Dsup, Drofu Dch Dimpl [TROi] [VDroiT] [COUNTRDY] 17	Dl, ^Dsup Drofu, Dch Dimpl [TROi] [VDroiT] [COUNTRDY] 17	Dl, Dsup Drofu [SETDIAGUC] [TRBi] [INITRB] [OWEDONE] 4	
			Dl, ^Dsup, Drofu Du, De, ^Dg Dimpl [TROi] [VDroiT] [COUNTRDY] [TREERESET] 17	Dl, ^Dsup Drofu Du, De, ^Dg Dimpl [TROi] [VDroiT] [COUNTRDY] [TREERESET] 17		
(Continued on sheet 75 of 98)			Dl, ^Dsup, Drofu Du, ^De, Dg Dimpl [TROi] [VDroiT] [COUNTRDY] [TREERESET] 17	Dl, ^Dsup Drofu Du, ^De, Dg Dimpl [TROi] [VDroiT] [COUNTRDY] [TREERESET] 17		

Table A.13/X.862 (sheet 74 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP- COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
						Dsup, ^Ddyn Drrec ^Do, ^Nch Np [VNopiINC] [VDopiT] [COUNTRDY] 20.2	
					Dsup ^Do, Nch Drrec, Np [TOPi] [VNopiINC] [VDopiT] [SAVEAAIDN] [COUNTRDY] 17	Dsup ^Do, Nch Drrec, Np [VNopiINC] [VDopiT] [SAVEAAIDN] [COUNTRDY] 20.2	
		^Dsup, Drofu Dch [TROi] [VDroiT] [COUNTRDY] 17	^Dsup, Drofu Dch [TROi] [VDroiT] [COUNTRDY] 17		^Dsup, Drofu Dch [TROi] [VDroiT] [COUNTRDY] 17	^Dsup, Drofu Dch [VDroiT] [COUNTRDY] 20.2	
		^Dsup, Drofu Du, De, ^Dg [TROi] [VDroiT] [COUNTRDY] [TREERESET] 17	^Dsup, Drofu Du, De, ^Dg [TROi] [VDroiT] [COUNTRDY] [TREERESET] 17		^Dsup, Drofu Du, De, ^Dg [TROi] [VDroiT] [COUNTRDY] [TREERESET] 17	^Dsup, Drofu Du, De, ^Dg [VDroiT] [COUNTRDY] 20.2	
		^Dsup, Drofu Du, ^De, Dg [TROi] [VDroiT] [COUNTRDY] [TREERESET] 17	^Dsup, Drofu Du, ^De, Dg [TROi] [VDroiT] [COUNTRDY] [TREERESET] 17		^Dsup, Drofu Du, ^De, Dg [TROi] [VDroiT] [COUNTRDY] [TREERESET] 17	^Dsup, Drofu Du, ^De, Dg [VDroiT] [COUNTRDY] [TREERESET] 20.2	

Table A.13/X.862 (sheet 75 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
C-NOCHANGE (result-not-required) ind *(Concluded 2 of 2)*			Dl, ^Dsup, Drofu Du, ^De, ^Dg Dimpl [DELBRO] [TROi] [DELBRANCH] [TREERESET] [CNCrsND] [COUNTRDY] [RESETD] 2			
				Dl, ^Dsup Drofu Du, ^De, ^Dg ^Dec, Dimpl [DELBRO] [TROi] [DELBRANCH] [TREERESET] [CNCrsND] [COUNTRDY] [RESETD] 3		
AF-EARLY-EXIT ind			^Dsup Deefu Dch [TEEi] [VDeeiT] [VDgF] [VDeF] [AEErs] [COUNTRDY] 17	^Dsup Deefu Dch [TEEi] [VDeeiT] [VDgF] [VDeF] [AEErs] [COUNTRDY] 17	^Dsup Deefu Dch [TEEi] [VDeeiT] [VDgF] [VDeF] [AEErs] [COUNTRDY] 17	^Dsup Deefu Dch [TEEi] [VDeeiT] [VDgF] [VDeF] [AEErs] [COUNTRDY] 17
			^Dsup Deefu ^Dch [DELBRO] [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBRANCH] [TREERESET] [AEErs] [COUNTRDY] [RESETD] 2			^Dsup Deefu ^Dch [DELBRO] [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBRANCH] [TREERESET] [AEErs] [COUNTRDY] [RESETD] 2
				^Dsup Deefu ^Dch [DELBRO] [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBRANCH] [TREERESET] [AEErs] [COUNTRDY] [RESETD] 3	^Dsup Deefu ^Dch [DELBRO] [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBRANCH] [TREERESET] [AEErs] [COUNTRDY] [RESETD] 3	

Table A.13/X.862 (sheet 76 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
		^Dsup, Drofu Du, ^De, ^Dg Dec [DELBRO] [TROi] [VDpsF] [DELBRANCH] [TREERESET] [CNCrsND] [COUNTRDY] [RESETD] 2	^Dsup, Drofu Du, ^De, ^Dg Dec [DELBRO] [TROi] [VDpsF] [DELBRANCH] [TREERESET] [CNCrsND] [COUNTRDY] [RESETD] 2		^Dsup, Drofu Du, ^De, ^Dg Dec [DELBRO] [TROi] [DELBRANCH] [TREERESET] [CNCrsND] [COUNTRDY] [RESETD] 2	^Dsup, Drofu Du, ^De, ^Dg Dec [DELBRO] [DELBRANCH] [TREERESET] [CNCrsND] [COUNTRDY] [RESETD] 2	
		^Dsup, Drofu Du, ^De, ^Dg ^Dec [DELBRO] [TROi] [VDpsF] [DELBRANCH] [TREERESET] [CNCrsND] [COUNTRDY] [RESETD] 3	^Dsup, Drofu Du, ^De, ^Dg ^Dec [DELBRO] [TROi] [VDpsF] [DELBRANCH] [TREERESET] [CNCrsND] [COUNTRDY] [RESETD] 3		^Dsup, Drofu Du, ^De, ^Dg ^Dec [DELBRO] [TROi] [DELBRANCH] [TREERESET] [CNCrsND] [COUNTRDY] [RESETD] 3	^Dsup, Drofu Du, ^De, ^Dg ^Dec [DELBRO] [DELBRANCH] [CNCrsND] [COUNTRDY] [RESETD] 3	
		^Dsup Deefu Dch [TEEi] [VDeeiT] [VDgF] [VDeF] [AEErs] [COUNTRDY] 17	^Dsup Deefu Dch [TEEi] [VDeeiT] [VDgF] [VDeF] [AEErs] [COUNTRDY] 17		^Dsup Deefu Dch [TEEi] [VDeeiT] [VDgF] [VDeF] [AEErs] [COUNTRDY] 17	^Dsup Deefu ^Du [SETDIAGEC] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	
		^Dsup Deefu ^Dch, Dec [DELBRO] [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBRANCH] [TREERESET] [AEErs] [COUNTRDY] [RESETD] 2	^Dsup Deefu ^Dch, Dec [DELBRO] [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBRANCH] [TREERESET] [AEErs] [COUNTRDY] [RESETD] 2		^Dsup Deefu ^Dch, Dec [DELBRO] [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBRANCH] [TREERESET] [AEErs] [COUNTRDY] [RESETD] 2	^Dsup Du Deefu [SETDIAGEC] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	
		^Dsup Deefu ^Dch, ^Dec [DELBRO] [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBRANCH] [TREERESET] [AEErs] [COUNTRDY] [RESETD] 3	^Dsup Deefu ^Dch, ^Dec [DELBRO] [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBRANCH] [TREERESET] [AEErs] [COUNTRDY] [RESETD] 3		^Dsup Deefu ^Dch, ^Dec [DELBRO] [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBRANCH] [TREERESET] [AEErs] [COUNTRDY] [RESETD] 3		

Table A.13/X.862 (sheet 77 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
TP-ROLLBACK req			Dl, Dsup ^Ncr, ^Nfrb [INITRB] [OWEDONE] [VDaT] [CANCEL] 23.3	Dl, Dsup ^Ncr, ^Nfrb [INITRB] [OWEDONE] [VDaT] [CANCEL] 23.3	Dl, Dsup ^Nfrb [INITRB] [OWEDONE] [CANCEL] 23.3	Dl, Dsup ^Ncr, ^Nfrb [INITRB] [OWEDONE] [VDaT] [CANCEL] 23.3
			Dl, Dsup ^Ncr, Nfrb [VDaT] [CANCEL] 23.3	Dl, Dsup ^Ncr, Nfrb [VDaT] [CANCEL] 23.3	Dl, Dsup Nfrb [CANCEL] 23.3	Dl, Dsup ^Ncr, Nfrb [VDaT] [CANCEL] 23.3
			Dl, ^Dsup ^Nfrb [RBREQ] [INITRB] [OWEDONE] 23.1	Dl, ^Dsup ^Nfrb [RBREQ] [INITRB] [OWEDONE] 23.1	Dl, ^Dsup ^Nfrb [RBREQ] [INITRB] [OWEDONE] 23.1	Dl, ^Dsup ^Nfrb [RBREQ] [INITRB] [OWEDONE] 23.1
			Nfrb, Dl, ^Dsup [RBREQ] 23.1	Nfrb, Dl, ^Dsup [RBREQ] 23.1	Nfrb, Dl, ^Dsup [RBREQ] 23.1	Nfrb, Dl, ^Dsup [RBREQ] 23.1
C-ROLLBACK ind			Dl, Dsup [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.4	Dl, Dsup [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.4	Dl, Dsup [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.4	Dl, Dsup [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.4
			^Du, Dl, ^Dsup ^Dcr [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Du, Dl, ^Dsup ^Dcr [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Du, Dl, ^Dsup ^Dcr [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Du, Dl, ^Dsup [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2
			Du, Dl, ^Dsup ^Dcr [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	Du, Dl, ^Dsup ^Dcr [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	Du, Dl, ^Dsup ^Dcr [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	Du, Dl, ^Dsup [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2

Table A.13/X.862 (sheet 78 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
		Dsup ^Nfrb [INITRB] [OWEDONE] [VDaT] [CANCEL] 23.3	Dsup ^Nfrb [INITRB] [OWEDONE] [VDaT] [CANCEL] 23.3	Dsup ^Nfrb [INITRB] [OWEDONE] [VDaT] [CANCEL] 23.3	Dsup ^Nfrb ^Ncr [INITRB] [OWEDONE] [VDaT] [CANCEL] 23.3		
		Dsup Nfrb [VDaT] [CANCEL] 23.3	Dsup Nfrb [VDaT] [CANCEL] 23.3	Dsup Nfrb [VDaT] [CANCEL] 23.3	Dsup Nfrb ^Ncr [VDaT] [CANCEL] 23.3		
		^Dsup ^Nfrb [RBREQ] [INITRB] [OWEDONE] 23.1	^Dsup ^Nfrb [RBREQ] [INITRB] [OWEDONE] 23.1	^Dsup ^Nfrb ^Deei [RBREQ] [INITRB] [OWEDONE] 23.1	^Dsup ^Nfrb [RBREQ] [INITRB] [OWEDONE] 23.1		
		^Dsup Nfrb [RBREQ] 23.1	^Dsup Nfrb [RBREQ] 23.1	^Dsup Nfrb ^Deei [RBREQ] 23.1	^Dsup Nfrb [RBREQ] 23.1		
				^Nfrb ^Dsup, Deei [INITRB] [OWEDONE] [COUNTRB] 23.2			
				Nfrb ^Dsup, Deei [COUNTRB] 23.2			
		Dsup [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.4	Dsup [SETDIAGSP] [TRBi] [LOGDAMRB] [INITRB] [OWEDONE] 23.4		Dsup [SETDIAGSP] [TRBi] [LOGDAMRB] [INITRB] [OWEDONE] 23.4	Dsup [SETDIAGSP] [TRBi] [LOGDAMRB] [INITRB] [OWEDONE] 23.4	
		^Dsup, ^Du ^Dcr [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, ^Du ^Dcr [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2		^Dsup, ^Du ^Dcr [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, ^Du [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	
		^Dsup, Du, ^Dcr [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	^Dsup, Du, ^Dcr [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2		^Dsup, Du, ^Dcr [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	^Dsup, Du [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	

Table A.13/X.862 (sheet 79 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates Event		Dsup		^Dsh		^Dsh
C-CANCEL ind			Dl, Dsup, Dch [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.3	Dl, Dsup, Dch [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.3	Dl, Dsup, Dch [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.3	Dl, Dsup, Dch [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.3
			Dl, ^Dsup [SETDIAGSB] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1	Dl, ^Dsup [SETDIAGSB] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1	Dl, ^Dsup [SETDIAGSB] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1	Dl, ^Dsup [SETDIAGSB] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1
AF-REPORT (rollbackRI, heuristic-report) ind			^Dsup ^Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup ^Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup ^Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup ^Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2
			^Dsup Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	^Dsup Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	^Dsup Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	^Dsup Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2
AF-REPORT (rollbackRI, heuristic-report, completion-report) ind			^Dsup ^Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2

(Continued on sheet 81 of 98)

Table A.13/X.862 (sheet 80 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
		Dsup, Dch [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.3	^Dsup [SETDIAGSB] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1		^Dsup [SETDIAGSB] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1	Dsup, Dch [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.3	
		^Dsup [SETDIAGSB] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1				^Dsup [SETDIAGSB] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1	
		^Dsup ^Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup ^Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2		^Dsup ^Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup ^Du ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	
		^Dsup Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	^Dsup Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2		^Dsup Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	^Dsup Du ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	
		^Dsup ^Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2		^Dsup ^Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Du ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	

Table A.13/X.862 (sheet 81 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
AF-REPORT (rollbackRI, heuristic-report, completion-report) ind *(Concluded 2 of 2)*			^Dsup Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	^Dsup Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	^Dsup Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	^Dsup Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2
AF-REPORT (rollbackRI, completion-report) ind			^Dsup ^Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2
			^Dsup Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	^Dsup Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	^Dsup Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	^Dsup Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2
AF-ABORT (provider, rollbackRI) ind			Du, Dl, ^Dsup ^Dbcr [TPABiR] [CRBrs] [SDETrqF] [ABDET] [INITRB] [OWEDONE] [COUNTRB] 23.2	Du, Dl, ^Dsup ^Dbcr [TPABiR] [CRBrs] [SDETrqF] [ABDET] [INITRB] [OWEDONE] [COUNTRB] 23.2	Du, Dl, ^Dsup ^Dbcr [TPABiR] [CRBrs] [SDETrqF] [ABDET] [INITRB] [OWEDONE] [COUNTRB] 23.2	Du, Dl, ^Dsup ^Dbcr [TPABiR] [CRBrs] [SDETrqF] [ABDET] [INITRB] [OWEDONE] [COUNTRB] 23.2

Table A.13/X.862 (sheet 82 of 98)

11	**12**	**15**	**16.1**	**17**	**18**	**20.1**	**20.2**
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
		^Dsup Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	^Dsup Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2		^Dsup Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	^Dsup Du ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	
		^Dsup ^Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2		^Dsup ^Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Du Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	
		^Dsup Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	^Dsup Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2		^Dsup Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	^Dsup Du Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	
		^Dsup Du ^Dbcr [TPABiR] [CRBrs] [SDETrqF] [ABDET] [INITRB] [OWEDONE] [COUNTRB] 23.2				^Dsup Du ^Dbcr [TPABiR] [CRBrs] [SDETrqF] [ABDET] [INITRB] [OWEDONE] [COUNTRB] 23.2	

Table A.13/X.862 (sheet 83 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates Event		Dsup		^Dsh		^Dsh
AF-ABORT (user, rollbackRI) ind			Dl, Dsup ^Ncr [TUABiR] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4	Dl, Dsup ^Ncr [TUABiR] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4	Dl, Dsup [TUABiR] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4	Dl, Dsup ^Ncr [TUABiR] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4
			Dl, Dsup Ncr [TUABi] [CRBrs] [SDETrqF] [DELBR] 1	Dl, Dsup Ncr [TUABi] [CRBrs] [SDETrqF] [DELBR] 1		Dl, Dsup Ncr [TUABi] [CRBrs] [SDETrqF] [DELBR] 1
			Dl, ^Dsup ^Dcr [TUABiR] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup ^Dcr [TUABiR] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup ^Dcr [TUABiR] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup [TUABiR] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2
AF-ABORT-AND-REPORT (rollbackRI, heuristic-report) ind			^Dsup ^Dhrsfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup ^Dhrsfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup ^Dhrsfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup ^Dhrsfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2
AF-ABORT-AND-REPORT (rollbackRI, heuristic-report, completion-report) ind			^Dsup ^Dhrsfu Dcdfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Dhrsfu Dcdfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Dhrsfu Dcdfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Dhrsfu Dcdfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2

Table A.13/X.862 (sheet 84 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP- COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
		Dsup [TUABiR] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4	Dsup [TUABiR] [LOGDAMRB] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4		Dsup [TUABiR] [LOGDAMRB] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4	Dsup [TUABiR] [LOGDAMRB] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4	
		^Dsup ^Dcr [TUABiR] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup [TUABiR] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2		^Dsup [TUABiR] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup [TUABiR] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	
		^Dsup ^Dcr ^Dhrsfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup ^Dhrsfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2		^Dsup ^Dhrsfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup ^Dhrsfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	
		^Dsup ^Dcr ^Dhrsfu Dcdfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Dhrsfu Dcdfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2		^Dsup ^Dhrsfu Dcdfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Dhrsfu Dcdfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	

Table A.13/X.862 (sheet 85 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
AF-ABORT-AND-REPORT (rollbackRI, completion-report) ind			^Dsup Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2
CAF-RECOVER (ready) ind			Dsup, Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dsup, Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dsup, Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dsup, Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8
			^Dsup, Drrec Dimpl [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, Drrec Dimpl [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, Drrec Dimpl [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, Drrec Dimpl [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2
Heuristic-decision						
Heuristic-damage-comp						

Table A.13/X.862 (sheet 86 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
		^Dsup ^Dcr Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2		^Dsup Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	
		Dsup, Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dsup, Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dsup Drdyi [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dsup, Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dsup, Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dsup Drdyi [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8
		^Dsup, Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup Drdyi [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup Drdyi [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2
			Dsup [LOGHD] 16.1	Dsup [LOGHD] 17	Dsup [LOGHD] 18	Dsup [LOGHD] 20.1	Dsup [LOGHD] 20.2
			Ni, ^Dsup 16.1	Ni, ^Dsup 17	Ni, ^Dsup 18	Ni, ^Dsup 20.1	Ni, ^Dsup 20.2
			memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 16.1	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 17	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 18	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 20.1	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 20.2

Table A.13/X.862 (sheet 87 of 98)

State	1	1.1	2	3	4	5
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
Rollback-by-TPPM			Dl, Dsup ^Ncr, ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3	Dl, Dsup ^Ncr, ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3	Dl, Dsup ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3	Dl, Dsup ^Ncr, ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3
			Dl, Dsup ^Ncr, Nfrb [CANCEL] 23.3	Dl, Dsup ^Ncr, Nfrb [CANCEL] 23.3	Dl, Dsup Nfrb [CANCEL] 23.3	Dl, Dsup ^Ncr, Nfrb [CANCEL] 23.3
			Dl, ^Dsup ^Nfrb [SETDIAGLO] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1	Dl, ^Dsup ^Nfrb [SETDIAGLO] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1	Dl, ^Dsup ^Nfrb [SETDIAGLO] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1	Dl, ^Dsup ^Nfrb [SETDIAGLO] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1
			Dl, ^Dsup Nfrb [RBREQ] 23.1	Dl, ^Dsup Nfrb [RBREQ] 23.1	Dl, ^Dsup Nfrb [RBREQ] 23.1	Dl, ^Dsup Nfrb [RBREQ] 23.1
Rollback-all			^Dsup [RBREQ] 23.1	^Dsup [RBREQ] 23.1	^Dsup [RBREQ] 23.1	^Dsup [RBREQ] 23.1
			Dsup [CANCEL] 23.3	Dsup [CANCEL] 23.3	Dsup [CANCEL] 23.3	Dsup [CANCEL] 23.3
send-prepare						

Table A.13/X.862 (sheet 88 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
		Dsup ^Nfrb ^Ncr [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3	Dsup ^Nfrb ^Ncr [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3	Dsup ^Nfrb ^Ncr [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3	Dsup ^Nfrb ^Ncr [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3	Dsup ^Nfrb ^Ncr [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3	Dsup ^Nfrb ^Ncr [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3
		Dsup Nfrb ^Ncr [CANCEL] 23.3	Dsup Nfrb ^Ncr [CANCEL] 23.3	Dsup Nfrb ^Ncr [CANCEL] 23.3	Dsup Nfrb ^Ncr [CANCEL] 23.3	Dsup Nfrb ^Ncr [CANCEL] 23.3	Dsup Nfrb ^Ncr [CANCEL] 23.3
		^Dsup ^Nfrb [SETDIAGLO] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1	^Dsup ^Nfrb [SETDIAGLO] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1	^Dsup, ^Deei ^Nfrb [SETDIAGLO] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1	^Dsup ^Nfrb [SETDIAGLO] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1	^Dsup ^Nfrb [SETDIAGLO] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1	^Dsup, ^Deei ^Nfrb [SETDIAGLO] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1
		^Dsup Nfrb [RBREQ] 23.1	^Dsup Nfrb [RBREQ] 23.1	^Dsup, ^Deei Nfrb [RBREQ] 23.1	^Dsup Nfrb [RBREQ] 23.1	^Dsup Nfrb [RBREQ] 23.1	^Dsup, ^Deei Nfrb [RBREQ] 23.1
				^Dsup, Deei ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [COUNTRB] 23.2			^Dsup, Deei ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [COUNTRB] 23.2
				^Dsup, Deei Nfrb [COUNTRB] 23.2			^Dsup, Deei Nfrb [COUNTRB] 23.2
		^Dsup [RBREQ] 23.1	^Dsup [RBREQ] 23.1	^Dsup, ^Deei [RBREQ] 23.1	^Dsup [RBREQ] 23.1	^Dsup [RBREQ] 23.1	^Dsup, ^Deei [RBREQ] 23.1
				^Dsup, Deei [COUNTRB] 23.2			^Dsup, Deei [COUNTRB] 23.2
		Dsup [CANCEL] 23.3	Dsup [CANCEL] 23.3	Dsup [CANCEL] 23.3	Dsup [CANCEL] 23.3	Dsup [CANCEL] 23.3	Dsup [CANCEL] 23.3
						Dps 20.1	20.02
						^Dps [GENPREP] 20.1	

Table A.13/X.862 (sheet 89 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
send-ready?						
one-ready						
Enter-ready-state						

Table A.13/X.862 (sheet 90 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
						Drsen, Drrec, Dps ^Ldready 20.1	20.2
						Drsen, Drrec, ^Dps ^Ldready [GENPREP] 20.1	
						Drsen, Drrec Ldready [SENDRDY?] 20.1	
						Drsen, ^Drrec [SENDRDY?] 20.1	
						^Drsen, Dps 20.1	
						^Drsen, ^Dps [GENPREP] 20.1	
							Drdyi Drsen ^Ldready [DECISION] 20.2
							Drdyi Drsen Ldready [SNDORDCD] 20.2
							Drdyi ^Drsen [DECISION] 20.2
							^Drdyi 20.2
						Dsup [DELIMIT] [SEND2PC] 20.3	20.3
						^Dsup [SEND2PC] 20.3	

Table A.13/X.862 (sheet 91 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
enter-one-phase-state						
enter-read-only-state						
enter-early-exit-state						
Set-done-true						
Continue-commit *(Continued on sheet 93 of 98)*						

Table A.13/X.862 (sheet 92 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
						Dsup Drsen [DELIMIT] [SEND1PC] 20.3.2	^Drdyi 20.3.2
						^Dsup [SEND1PC] 20.3.2	Drdyi, Dsup [DELIMIT] [CRDYRESET] [VNrdyiDEC] [VDrdyiF] [SEND1PC] 20.3.2
							Drdyi, ^Dsup [CRDYRESET] [VNrdyiDEC] [VDrdyiF] [SEND1PC] 20.3.2
						Dsup [DELIMIT] [SENDRO] 20.3.3	^Drdyi 20.3.3
							Dsup, Drdyi [DELIMIT] [CRDYRESET] [VNrdyiDEC] [VDrdyiF] [SENDRO] 20.3.3
						Dsup, Neer [DELIMIT] [SENDEE] 20.3.3	20.3.3
						[VDdT] 20.1	[VDdT] 20.2
							^Dsup, Drdyi ^De, Ptok [COMREQ] 21.1
							^Dsup, Drdyi De, Ptok [NOTCHAIN] [COMREQ] 21.1
							^Dsup, Dopi, Du ^Db, ^De, ^Dtb [COMREQ] [CPSAP] 21.3
							^Dsup, Dopi, Du ^Db, ^De, Dtb [COMREQ] [SDETrqF] [ABDET] 21.3

Table A.13/X.862 (sheet 93 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
Continue-commit *(Continued 2 of 4)* *(Continued on sheet 95 of 98)*						

Table A.13/X.862 (sheet 94 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP- COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
							^Dsup, Dopi, Du ^Db, De [COMREQ] [SDETrqF] [ABDET] 21.3
							^Dsup, Droi, ^Db ^Dch, ^De, ^Dtb [COMREQ] [CPSAP] 21.3
							^Dsup, Droi, ^Db De [NOTCHAIN] [COMREQ] [SDETrqF] [ABDET] 21.3
							^Dsup, Droi, ^Db Dch, ^De, ^Dtb 21.3
							^Dsup, Droi, Db 21.3
							^Dsup, Deei 21.3
							Dsup, Drdyi ^Db, ^De Ptok Dhrsfu, ^Dcdfu [COMREQ] 21.5.2
							Dsup, Drdyi ^Db, ^De Ptok Dhrsfu, Dcdfu [COMREQ] [INITREPSP] 21.5.1
							Dsup, Drdyi ^Db, ^De Ptok ^Dhrsfu, ^Dcdfu [COMREQ] [INITREPSP] 21.5.1
							Dsup, Drdyi ^Db, ^De Ptok ^Dhrsfu, Dcdfu [COMREQ] [INITREPSP] 21.5.1
							Dsup, Drdyi ^Db, De Ptok Dhrsfu, ^Dcdfu [NOTCHAIN] [COMREQ] 21.5.2

Table A.13/X.862 (sheet 95 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
Continue-commit *(Continued 3 of 4)* *(Continued on sheet 97 of 98)*						

Table A.13/X.862 (sheet 96 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END-DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END-DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
							Dsup, Drdyi ^Db, De Ptok Dhrsfu, Dcdfu [NOTCHAIN] [COMREQ] [INITREPSP] 21.5.1
							Dsup, Drdyi ^Db, De Ptok ^Dhrsfu, ^Dcdfu [NOTCHAIN] [COMREQ] [INITREPSP] 21.5.1
							Dsup, Drdyi ^Db, De Ptok ^Dhrsfu, Dcdfu [NOTCHAIN] [COMREQ] [INITREPSP] 21.5.1
							Dsup, Dopi, ^Db ^Dch, ^De, ^Dtb Dhrsfu, ^Dcdfu [COMREQ] [CPSAP] 21.5.4
							Dsup, Dopi, ^Db ^Dch, ^De, ^Dtb Dhrsfu, Dcdfu [COMREQ] [INITREPSP] 21.5.1
							Dsup, Dopi, ^Db ^Dch, ^De, ^Dtb ^Dhrsfu, ^Dcdfu [COMREQ] [INITREPSP] 21.5.1
							Dsup, Dopi, ^Db ^Dch, ^De, ^Dtb ^Dhrsfu, Dcdfu [COMREQ] [INITREPSP] 21.5.1
							Dsup, Dopi, ^Db De Dhrsfu, ^Dcdfu [NOTCHAIN] [COMREQ] [SDETrqF] [ABDET] 21.5.4
							Dsup, Dopi, ^Db De Dhrsfu, Dcdfu [NOTCHAIN] [COMREQ] [INITREPSP] 21.5.1

Table A.13/X.862 (sheet 97 of 98)

State	**1**	**1.1**	**2**	**3**	**4**	**5**
	Dialogue does not exist	C-BEGIN ind awaited	Data transfer S.C. or P.C. w/ control	Data transfer P.C. w/o control	AF-U-ERROR req issued S.C. or P.C. w/o control	AF-U-ERROR ind received P.C. w/ control
Predicates **Event**		Dsup		^Dsh		^Dsh
Continue-commit *(Concluded 4 of 4)*						
activate-nfsm			2	3		
deactivate-nfsm			2	3		

Table A.13/X.862 (sheet 98 of 98)

11	12	15	16.1	17	18	20.1	20.2
AF-END- DIALOGUE (conf=TRUE) req issued S.C. or P.C. w/ control	AF-END- DIALOGUE (conf=TRUE) ind received S.C. or P.C. w/o control	TP-PREPARE req issued ready awaited S.C. or P.C. w/ control	TP-PREPARE req issued TP-PREPARE ind received	ready-signal received TP-COMMIT or substitute req awaited	C-PREPARE ind received	TP-COMMITreq or sub received ready-signal not received S.C. or P.C. w/ control	Last ready awaited ready-signal received, sync. or p-abort awaited
(^Dcr or ^Ncr), ^Dl	(^Dcr or ^Ncr), ^Dl	Dl, ((Dsup, Ddyn) or ^Dsup)	Dl (Ddyn or ^Do)	Dl, (^Dsup or (Dsup, (Ddyn or ^Do)))	Dl	^Dcr, Dl	Dl
							Dsup, Dopi, ^Db De ^Dhrsfu, Dcdfu [NOTCHAIN] [COMREQ] [INITREPSP] 21.5.1
							Dsup, Dopi, ^Db De ^Dhrsfu, ^Dcdfu [NOTCHAIN] [COMREQ] [INITREPSP] 21.5.1
							Dsup, Dopi, ^Db Dch, ^De, ^Dtb Dhrsfu, ^Dcdfu [COMREQ] 21.5.3
							Dsup, Dopi, ^Db Dch, ^De, ^Dtb Dhrsfu, Dcdfu [COMREQ] [INITREPSP] 21.5.1
							Dsup, Dopi, ^Db Dch, ^De, ^Dtb ^Dhrsfu, ^Dcdfu [COMREQ] [INITREPSP] 21.5.1
							Dsup, Dopi, ^Db Dch, ^De, ^Dtb ^Dhrsfu, Dcdfu [COMREQ] [INITREPSP] 21.5.1

End of Table A.13

Table A.14/X.862 – Handshake (sheet 1 of 20)

State	6	7	8	9	10	13	14
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates Event	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
TP-BEGIN-DIALOGUE (accepted) rsp		Dsup Ncr [DELIMIT] 7					Dsup Ncr [DELIMIT] 14
TP-BEGIN-DIALOGUE (rejected) rsp		^Du, ^Dl, Dsup ^Nrn, ^Da [ABDrsRUd] [SDETrqF] 1					^Du, ^Dl, Dsup ^Nrn, ^Da [ABDrsRUd] [SDETrqF] 1
		Du, ^Dl, Dsup ^Nrn, ^Da [ABDrsRUd] [SDETrqBF] 1					Du, ^Dl, Dsup ^Nrn, ^Da [ABDrsRUd] [SDETrqBF] 1
		Dl, Dsup ^Nrn, ^Da [ABDrsRUr] [SDETrqRBC] [REJTRAN] [TREERESET] 1					Dl, Dsup ^Nrn, ^Da [ABDrsRUr] [SDETrqRBC] [REJTRAN] [TREERESET] 1
AF-BEGIN-DIALOGUE (accepted, dataRI) cnf	^Dsup Dcr [TBDcX] [VDcrF] [VDaT] 6					^Dsup Dcr [TBDcX] [VDcrF] [VDaT] 13	
	^Dsup ^Dcr [VDaT] 6					^Dsup ^Dcr [VDaT] 13	
AF-BEGIN-DIALOGUE (rejected(provider), dataRI) cnf	^Dl, ^Dsup [TBDcX] [SDETrqF] 1					^Dl, ^Dsup [TBDcX] [SDETrqF] 1	
	Dl, ^Dsup [TBDcX] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25					Dl, ^Dsup [TBDcX] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25	

Table A.14/X.862 – Handshake (sheet 2 of 20)

State	**6**	**7**	**8**	**9**	**10**	**13**	**14**
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates **Event**	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
AF-BEGIN-DIALOGUE (rejected(user), dataRI) cnf	^Dl, ^Dsup [TBDcX] [SDETrqF] 1					^Dl, ^Dsup [TBDcX] [SDETrqF] 1	
	Du, Dl, ^Dsup [TBDcX] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25					Du, Dl, ^Dsup [TBDcX] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25	
AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) cnf	Dl, ^Dsup [TBDcX] [CRBrs] [SDETrqF] [ABDET] [DELBRANCH] [TREERESET] 25					Dl, ^Dsup [TBDcX] [CRBrs] [SDETrqF] [ABDET] [DELBRANCH] [TREERESET] 25	
SAF-ASSOCIATION-LOST ind	^Dl [TBDcRP] 1					^Dl [TBDcRP] 1	
	Dl [TBDcRP] [ABDET] [DELBRANCH] [TREERESET] 25					Dl [TBDcRP] [ABDET] [DELBRANCH] [TREERESET] 25	

Table A.14/X.862 – Handshake (sheet 3 of 20)

State	6	7	8	9	10	13	14
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates Event	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
AF-END-DIALOGUE (confirmation = FALSE) ind	Dsh, ^Dl, Dsup [TEDi] [SDETrqF] 1						
	Dsh, ^Dl, ^Dsup ^Dcr [TEDi] [SDETrqF] 1						
	Dsh, Dx [TPABiBTED] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25						
AF-END-DIALOGUE (confirmation = TRUE) ind	Dsh, ^Dl, Dsup Denb=0 [TEDi] 9						
	Dsh, ^Dl, Dsup Denb>0 [DECDENB] 6						
	Dsh, ^Dl, ^Dsup ^Dcr Denb=0 [TEDi] 9						
	Dsh, ^Dl, ^Dsup ^Dcr Denb>0 [DECDENB] 6						

(Continued on sheet 4 of 20)

Table A.14/X.862 – Handshake (sheet 4 of 20)

State	**6**	**7**	**8**	**9**	**10**	**13**	**14**
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates **Event**	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
AF-END-DIALOGUE (confirmation = TRUE) ind *(Concluded 2 of 2)*	Dsh, Dx Denbb=0 [TPABiBTED] [SDETrqRBR] [ABDET] [DELBRANCH] [TREERESET] 25						
	Dsh, Dx Denbb>0 [DECDENB] 6						
TP-U-ERROR req	Dsh, Dsup [AUErq] [VDenbINC] 6	Dsh, Dsup ^Ncr [DELIMIT] [AUErq] 2	[AUErq] 6	[AUErq] 6	[AUErq] 11		
	Dsh, ^Dsup [AUErq] [VDenbINC] 6	Dsh, ^Dsup [AUErq] 2					
		^Dsh, Dsup ^Ncr [DELIMIT] [AUErq] [VDecT] 2					Dsup ^Ncr [DELIMIT] [AUErq] [VDecT] 2
		^Dsh, ^Dsup [AUErq] [VDecT] 2					^Dsup [AUErq] [VDecT] 2
AF-U-ERROR ind	Dsh, Dsup Denb=0 [TUEi] 2	Dsh ^Da [TUEi] [VDepnbINC] 7					
(Continued on sheet 5 of 20)	Dsh, Dsup Denb>0 [TUEi] 4						

Table A.14/X.862 – Handshake (sheet 5 of 20)

State	**6**	**7**	**8**	**9**	**10**	**13**	**14**
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates **Event**	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
AF-U-ERROR ind *(Concluded 2 of 2)*	Dsh, ^Dsup ^Dcr Denb=0 [TUEi] 2	Dsh Da [TUEi] [AUErs] 7	[TUEi] 7	[TUEi] 12	[TUEi] 7	Dsup [TUEi] [VDecF] 3	
	Dsh, ^Dsup ^Dcr Denb>0 [TUEi] 4						
	^Dsh, Dsup [TUEi] [VDecF] 3						
	^Dsh, ^Dsup ^Dcr [TUEi] [VDecF] 3					^Dsup ^Dcr [TUEi] [VDecF] 3	
AF-U-ERROR cnf	Dsh, Denb>0 [DECDENB] 6						
TP-U-ABORT req	^Du, ^Dl, Dsup [AABrqUd] [SDETrqF] 1	^Du, ^Dl, Dsup ^Ncr [DELIMIT] [AABrqUd] [SDETrqF] 1	^Du, ^Dl, Dsup [AABrqUd] [SDETrqF] 1	^Du, Dsup [AABrqUd] [SDETrqF] 1	^Du, Dsup [AABrqUd] [SDETrqF] 1	^Du, ^Dl, Dsup [AABrqUd] [SDETrqF] 1	^Du, ^Dl, Dsup ^Ncr [DELIMIT] [AABrqUd] [SDETrqF] 1
	Du, ^Dl, Dsup [AABrqUd] [SDETrqBF] 1	Du, ^Dl, Dsup ^Ncr [DELIMIT] [AABrqUd] [SDETrqBF] 1	Du, ^Dl, Dsup [AABrqUd] [SDETrqBF] 1	Du, Dsup [AABrqUd] [SDETrqBF] 1	Du, Dsup [AABrqUd] [SDETrqBF] 1	Du, ^Dl, Dsup [AABrqUd] [SDETrqBF] 1	Du, ^Dl, Dsup ^Ncr [DELIMIT] [AABrqUd] [SDETrqBF] 1
(Continued on sheet 6 of 20)	^Dl, ^Dsup [AABrqUd] [SDETrqF] 1	^Dl, ^Dsup [AABrqUd] [SDETrqF] 1	^Dl, ^Dsup [AABrqUd] [SDETrqF] 1	^Dsup [AABrqUd] [SDETrqF] 1	^Dsup [AABrqUd] [SDETrqF] 1	^Dl, ^Dsup [AABrqUd] [SDETrqF] 1	^Dl, ^Dsup [AABrqUd] [SDETrqF] 1

Table A.14/X.862 – Handshake (sheet 6 of 20)

State	6	7	8	9	10	13	14
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates Event	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
TP-U-ABORT req *(Concluded 2 of 2)*	Dl, Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 6	Dl, Dsup ^Ncr [DELIMIT] [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 7	Dl, Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 8			Dl, Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 13	Dl, Dsup ^Ncr [DELIMIT] [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 14
	Dl, ^Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 6	Dl, ^Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 7	Dl, ^Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 8			Dl, ^Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 13	Dl, ^Dsup [ABTPSUI] [NOTCHAIN] [INITRB] [OWEDONE] 14
AF-ABORT (user, dataRI) ind	^Dl, Dsup [TUABi] [SDETrqF] 1	^Dl [TUABi] [SDETrqF] 1	^Dl [TUABi] [SDETrqF] 1	[TUABi] [SDETrqF] 1	[TUABi] [SDETrqF] 1	^Dl, Dsup [TUABi] [SDETrqF] 1	^Dl [TUABi] [SDETrqF] 1
	^Dl, ^Dsup ^Dcr [TUABi] [SDETrqF] 1					^Dl, ^Dsup ^Dcr [TUABi] [SDETrqF] 1	
	Dl, ^Dsup ^Dcr, ^Dbcr [TUABi] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25	Du, Dl, ^Dsup ^Dbcr [TUABi] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25	Du, Dl, ^Dsup ^Dbcr [TUABi] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25			Dl, ^Dsup ^Dcr, ^Dbcr [TUABi] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25	Du, Dl, ^Dsup ^Dbcr [TUABi] [SDETrqRB] [ABDET] [DELBRANCH] [TREERESET] 25

Table A.14/X.862 – Handshake (sheet 7 of 20)

State	**6**	**7**	**8**	**9**	**10**	**13**	**14**
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates **Event**	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
AF-ABORT (provider, abortRI) ind or A-ABORT ind or A-ABORT req or A-P-ABORT ind or A-RELEASE (result = affirmative) rsp or A-RELEASE (result = affirmative) cnf	^Dl [SETDIAGTP] [TPABi] 1	^Dl [SETDIAGTP] [TPABi] 1	^Dl [SETDIAGTP] [TPABi] 1	[SETDIAGTP] [TPABi] 1	[SETDIAGTP] [TPABi] 1	^Dl [SETDIAGTP] [TPABi] 1	^Dl [SETDIAGTP] [TPABi] 1
		Dl, Dsup Ncr [SETDIAGTP] [TPABi] [REJTRAN] [TREERESET] 1					Dl, Dsup Ncr [SETDIAGTP] [TPABi] [REJTRAN] [TREERESET] 1
	Dl, Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dl, Dsup ^Ncr [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dl, Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8			Dl, Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dl, Dsup ^Ncr [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8
	Dl, ^Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2			Dl, ^Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup [SETDIAGTP] [TPABiR] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2

Table A.14/X.862 – Handshake (sheet 8 of 20)

State	6	7	8	9	10	13	14
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates Event	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
Protocol error or Internal error	^Dl [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] 1	^Dl [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] 1	^Dl [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] 1	[SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] 1	[SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] 1	^Dl [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] 1	^Dl [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] 1
		Dl, Dsup Ncr [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [REJTRAN] [TREERESET] 1					Dl, Dsup Ncr [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [REJTRAN] [TREERESET] 1
	Dl, Dsup [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dl, Dsup ^Ncr [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dl, Dsup [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8			Dl, Dsup [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8	Dl, Dsup ^Ncr [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.8
	Dl, ^Dsup [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2			Dl, ^Dsup [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup [SETDIAGTP] [TPABiR] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2

Table A.14/X.862 – Handshake (sheet 9 of 20)

State	**6**	**7**	**8**	**9**	**10**	**13**	**14**
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates **Event**	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
TP-REQUEST-CONTROL req		^Dsh, Dsup ^Ncr [DELIMIT] [ARCrq] 7					
		^Dsh, ^Dsup [ARCrq] 7					
AF-REQUEST-CONTROL ind	^Dsh [TRCi] 6					13	
TP-HANDSHAKE req		Dsh, Dsup ^Ncr [DELIMIT] [AHSrq] 8					
		Dsh, ^Dsup [AHSrq] 8					
AF-HANDSHAKE ind	Dsh, Dsup Denb=0 [THSi] 8						
	Dsh, Dsup Denb>0 [DECDENB] 6						
	Dsh, ^Dsup ^Dcr Denb=0 [THSi] 8						
	Dsh, ^Dsup ^Dcr Denb>0 [DECDENB] 6						

Table A.14/X.862 – Handshake (sheet 10 of 20)

State	6	7	8	9	10	13	14
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates Event	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
TP-HANDSHAKE rsp		Dsh, Dsup ^Ncr [DELIMIT] [AHSrs] 2	[AHSrs] 6		[AHSrs] 11		
		Dsh, ^Dsup [AHSrs] 2					
		^Dsh, Dsup ^Ncr [DELIMIT] [AHSrs] [VDecF] 3					
		^Dsh, ^Dsup [AHSrs] [VDecF] 3					
AF-HANDSHAKE cnf	Dsh Denb>0 [THSc] 4		[THSc] 7	[THSc] 12			
	Dsh Denb=0 [THSc] 2						
	^Dsh [THSc] [VDecT] 2						
TP-HANDSHAKE-AND-GRANT-CONTROL rsp							Dsup ^Ncr [DELIMIT] [AHSGCrs] 2
							^Dsup [AHSGCrs] 2
AF-HANDSHAKE-AND-GRANT-CONTROL cnf						[THSGCc] [VDecF] 3	

Table A.14/X.862 – Handshake (sheet 11 of 20)

State	6	7	8	9	10	13	14
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates Event	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
C-BEGIN ind	Du, Dsh, ^Dl, Dsup Nr [TPABiBTR] [AABrqPrTR] [SDETrqRBC] 1						
	Du, Dsh, ^Dl, Dsup ^Nr Dgrp=2 [VDgrp3] [TREESET] [ADDBRSP] [TBTi] 6						
	Du, Dsh, ^Dl, Dsup ^Nr Dgrp=4 [TREESET] [ADDBRSP] [TBTi] 6						
AF-BEGIN-TRANSACTION ind	Du, Dsh, ^Dl, Dsup Nr [TPABiBTR] [AABrqPrTR] [SDETrqRBC] 1						
	Du, Dsh, ^Dl, Dsup ^Nr Dgrp=2 [VDgrp1] [TREESET] [TBDiSAVE] [ADDBRSP] [VNtpsuiT] 2						
C-BEGIN cnf	Dl, ^Dsup [VDbcrT] [VDxF] 6	Dl, ^Dsup [VDbcrT] [VDxF] 7	Dl, ^Dsup [VDbcrT] [VDxF] 8			Dl, ^Dsup [VDbcrT] [VDxF] 13	

Table A.14/X.862 – Handshake (sheet 12 of 20)

State	6	7	8	9	10	13	14
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates **Event**	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
TP-DATA req		Dsh, Dsup ^Ncr [DELIMIT] [UASErq] 7					
		Dsh, ^Dsup [UASErq] 7					
U-ASE ind	Dsh, Dsup Denb=0 [TDTi] 6						
	Dsh ^Dsup Denb=0 [TDTi] 6						
	Dsh Denb>0 6						
AF-DEFER (end-dialogue) ind	Dsh, Dl, Dsup ^De [TDEi] [VDeT] 6						
AF-PREPARE ind or C-READY ind or C-NOCHANGE (result-required) ind or C-NOCHANGE (result-not-required) ind or AF-NOCHANGE (result-required) ind	Dsh, Dl [SETDIAGUC] [TRBi] [INITRB] [OWEDONE] 6						
TP-ROLLBACK req	Dl, Dsup ^Nfrb [INITRB] [OWEDONE] 23.3	Dl, Dsup ^Ncr, ^Nfrb [INITRB] [OWEDONE] [VDaT] 23.3	Dl, Dsup ^Nfrb [INITRB] [OWEDONE] 23.3			Dl, Dsup ^Nfrb [INITRB] [OWEDONE] 23.3	Dl, Dsup ^Ncr, ^Nfrb [INITRB] [OWEDONE] [VDaT] 23.3
	Dl, Dsup Nfrb 23.3	Dl, Dsup ^Ncr, Nfrb [VDaT] 23.3	Dl, Dsup Nfrb 23.3			Dl, Dsup Nfrb 23.3	Dl, Dsup ^Ncr, Nfrb [VDaT] 23.3

(Continued on sheet 13 of 20)

Table A.14/X.862 – Handshake (sheet 13 of 20)

State	**6**	**7**	**8**	**9**	**10**	**13**	**14**
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates **Event**	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
TP-ROLLBACK req *(Concluded 2 of 2)*	Dl, ^Dsup ^Nfrb [RBREQ] [INITRB] [OWEDONE] 23.1	Dl, ^Dsup ^Nfrb [RBREQ] [INITRB] [OWEDONE] 23.1	Dl, ^Dsup ^Nfrb [RBREQ] [INITRB] [OWEDONE] 23.1			Dl, ^Dsup ^Nfrb [RBREQ] [INITRB] [OWEDONE] 23.1	Dl, ^Dsup ^Nfrb [RBREQ] [INITRB] [OWEDONE] 23.1
	Dl, ^Dsup Nfrb [RBREQ] 23.1	Dl, ^Dsup Nfrb [RBREQ] 23.1	Dl, ^Dsup Nfrb [RBREQ] 23.1			Dl, ^Dsup Nfrb [RBREQ] 23.1	Dl, ^Dsup Nfrb [RBREQ] 23.1
C-ROLLBACK ind	Dl, Dsup [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.4	Dl, Dsup [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.4	Dl, Dsup [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.4			Dl, Dsup [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.4	Dl, Dsup [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.4
	^Du, Dl, ^Dsup ^Dcr [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Du, Dl, ^Dsup [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Du, Dl, ^Dsup [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2			^Du, Dl, ^Dsup ^Dcr [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Du, Dl, ^Dsup [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2
	Du, Dl, ^Dsup ^Dcr [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	Du, Dl, ^Dsup [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	Du, Dl, ^Dsup [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2			Du, Dl, ^Dsup ^Dcr [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	Du, Dl, ^Dsup [SETDIAGSB] [TRBi] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2

Table A.14/X.862 – Handshake (sheet 14 of 20)

State	6	7	8	9	10	13	14
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates Event	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
C-CANCEL ind	Dl, Dsup, Dch [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.3						
	Dl, ^Dsup [SETDIAGSB] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1	Dl, ^Dsup [SETDIAGSB] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1	Dl, ^Dsup [SETDIAGSB] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1			Dl, ^Dsup [SETDIAGSB] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1	Dl, ^Dsup [SETDIAGSB] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1
AF-REPORT (rollbackRI, heuristic-report) ind	^Dsup ^Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup ^Du ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup ^Du ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2			^Dsup ^Du ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup ^Du ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2
	^Dsup Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	^Dsup Du ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	^Dsup Du ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2			^Dsup Du ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	^Dsup Du ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2

Table A.14/X.862 – Handshake (sheet 15 of 20)

State	6	7	8	9	10	13	14
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates Event	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
AF-REPORT (rollbackRI, heuristic-report, completion-report) ind	^Dsup ^Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Du ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Du ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2			^Dsup ^Du ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Du ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2
	^Dsup Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	^Dsup Du ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	^Dsup Du ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2			^Dsup Du ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	^Dsup Du ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2

Table A.14/X.862 – Handshake (sheet 16 of 20)

State	**6**	**7**	**8**	**9**	**10**	**13**	**14**
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates **Event**	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
AF-REPORT (rollbackRI, completion-report) ind	^Dsup ^Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Du Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Du Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2			^Dsup ^Du Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Du Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] 23.2
	^Dsup Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	^Dsup Du Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	^Dsup Du Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2			^Dsup Du Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2	^Dsup Du Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [CRALL] [COUNTRB] [CPSAP] 23.2
AF-EARLY-EXIT ind *(Continued on sheet 17 of 20)*	^Dsup Deefu Dch [TEEi] [VDeeiT] [VDgF] [VDeF] [VNcntrdyDEC] [AEErs] 17	^Dsup Deefu Dch [TEEi] [VDeeiT] [VDgF] [VDeF] [VNcntrdyDEC] [AEErs] 17	^Dsup Deefu Dch [TEEi] [VDeeiT] [VDgF] [VDeF] [VNcntrdyDEC] [AEErs] 17			^Dsup Deefu Dch [TEEi] [VDeeiT] [VDgF] [VDeF] [VNcntrdyDEC] [AEErs] 17	^Dsup Deefu Dch [TEEi] [VDeeiT] [VDgF] [VDeF] [VNcntrdyDEC] [AEErs] 17

Table A.14/X.862 – Handshake (sheet 17 of 20)

State	**6**	**7**	**8**	**9**	**10**	**13**	**14**
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates **Event**	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
AF-EARLY-EXIT ind *(Concluded 2 of 2)*	^Dsup Deefu ^Dch, Dec [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBR] [DELBRANCH] [TREERESET] [AEErs] 2	^Dsup Deefu ^Dch, Dec [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBR] [DELBRANCH] [TREERESET] [AEErs] 2	^Dsup Deefu ^Dch, Dec [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBR] [DELBRANCH] [TREERESET] [AEErs] 2			^Dsup Deefu ^Dch, Dec [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBR] [DELBRANCH] [TREERESET] [AEErs] 2	^Dsup Deefu ^Dch, Dec [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBR] [DELBRANCH] [TREERESET] [AEErs] 2
	^Dsup Deefu ^Dch, ^Dec [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBR] [DELBRANCH] [TREERESET] [AEErs] 3	^Dsup Deefu ^Dch, ^Dec [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBR] [DELBRANCH] [TREERESET] [AEErs] 3	^Dsup Deefu ^Dch, ^Dec [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBR] [DELBRANCH] [TREERESET] [AEErs] 3			^Dsup Deefu ^Dch, ^Dec [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBR] [DELBRANCH] [TREERESET] [AEErs] 3	^Dsup Deefu ^Dch, ^Dec [TEEi] [VDeeiT] [VDgF] [VDeF] [DELBR] [DELBRANCH] [TREERESET] [AEErs] 3
AF-ABORT (provider, rollbackRI) ind	Du, Dl, ^Dsup ^Dbcr [TPABiR] [CRBrs] [SDETrqF] [ABDET] [INITRB] [OWEDONE] [COUNTRB] 23.2	Du, Dl, ^Dsup ^Dbcr [TPABiR] [CRBrs] [SDETrqF] [ABDET] [INITRB] [OWEDONE] [COUNTRB] 23.2	Du, Dl, ^Dsup ^Dbcr [TPABiR] [CRBrs] [SDETrqF] [ABDET] [INITRB] [OWEDONE] [COUNTRB] 23.2			Du, Dl, ^Dsup ^Dbcr [TPABiR] [CRBrs] [SDETrqF] [ABDET] [INITRB] [OWEDONE] [COUNTRB] 23.2	

Table A.14/X.862 – Handshake (sheet 18 of 20)

State	**6**	**7**	**8**	**9**	**10**	**13**	**14**
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates **Event**	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
AF-ABORT (user, rollbackRI) ind	Dl, Dsup [TUABiR] [INITRB] [OWEDONE] [ABPTNR] [NOTCHAIN] 23.4	Dl, Dsup ^Ncr [TUABiR] [INITRB] [OWEDONE] [ABPTNR] [NOTCHAIN] 23.4	Dl, Dsup [TUABiR] [INITRB] [OWEDONE] [ABPTNR] [NOTCHAIN] 23.4			Dl, Dsup [TUABiR] [INITRB] [OWEDONE] [ABPTNR] [NOTCHAIN] 23.4	Dl, Dsup ^Ncr [TUABiR] [INITRB] [OWEDONE] [ABPTNR] [NOTCHAIN] 23.4
		Dl, Dsup Ncr [TUABi] [CRBrs] [SDETrqF] [DELBR] 1					Dl, Dsup Ncr [TUABi] [CRBrs] [SDETrqF] [DELBR] 1
	Dl, ^Dsup ^Dcr [TUABiR] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup [TUABiR] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup [TUABiR] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2			Dl, ^Dsup ^Dcr [TUABiR] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	Dl, ^Dsup [TUABiR] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2
AF-ABORT-AND-REPORT (rollbackRI, heuristic-report) ind	^Dsup ^Dcr ^Dhrsfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup ^Dhrsfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup ^Dhrsfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2			^Dsup ^Dhrsfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup ^Dhrsfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2

Table A.14/X.862 – Handshake (sheet 19 of 20)

State	6	7	8	9	10	13	14
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates Event	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
AF-ABORT-AND-REPORT (rollbackRI, heuristic-report, completion-report) ind	^Dsup ^Dcr ^Dhrsfu Dcdfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Dhrsfu Dcdfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Dhrsfu Dcdfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2			^Dsup ^Dhrsfu Dcdfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup ^Dhrsfu Dcdfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2
AF-ABORT-AND-REPORT (rollbackRI, completion-report) ind	^Dsup ^Dcr Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2			^Dsup Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2	^Dsup Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [CRALL] [COUNTRB] 23.2

Table A.14/X.862 – Handshake (sheet 20 of 20)

State	**6**	**7**	**8**	**9**	**10**	**13**	**14**
	AF-HANDSHAKE req issued	AF-HANDSHAKE ind received	AF-HANDSHAKE ind received on state 6, or req issued on state 7	AF-END-DIALOGUE ind received on state 6	AF-HANDSHAKE ind received on state 11	AF-HANDSHAKE &-GRANT-CTL req issued	AF-HANDSHAKE &-GRANT-CTL ind received
Predicates **Event**	P.C. or S.C.	P.C. or S.C.	S.C.	S.C. ^Dl	S.C. ^Dl	P.C.	P.C.
Rollback-by-TPPM	Dl, Dsup ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] 23.3	Dl, Dsup ^Ncr, ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] 23.3	Dl, Dsup ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] 23.3			Dl, Dsup ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] 23.3	Dl, Dsup ^Ncr, ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] 23.3
	Dl, Dsup Nfrb 23.3	Dl, Dsup ^Ncr, Nfrb 23.3	Dl, Dsup Nfrb 23.3			Dl, Dsup Nfrb 23.3	Dl, Dsup ^Ncr, Nfrb 23.3
	Dl, ^Dsup ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [RBREQ] 23.1	Dl, ^Dsup ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [RBREQ] 23.1	Dl, ^Dsup ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [RBREQ] 23.1			Dl, ^Dsup ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [RBREQ] 23.1	Dl, ^Dsup ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [RBREQ] 23.1
	Dl, ^Dsup Nfrb [RBREQ] 23.1	Dl, ^Dsup Nfrb [RBREQ] 23.1	Dl, ^Dsup Nfrb [RBREQ] 23.1			Dl, ^Dsup Nfrb [RBREQ] 23.1	Dl, ^Dsup Nfrb [RBREQ] 23.1
Rollback-all	Dsup [CANCEL] 23.3	Dsup [CANCEL] 23.3	Dsup [CANCEL] 23.3			Dsup [CANCEL] 23.3	Dsup [CANCEL] 23.3
	^Dsup [RBREQ] 23.1	^Dsup [RBREQ] 23.1	^Dsup [RBREQ] 23.1			^Dsup [RBREQ] 23.1	^Dsup [RBREQ] 23.1

End of Table A.14

Table A.15-1/X.862 – Commitment (sheet 1 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
TP-U-ABORT req	^Danyb, Nfa Dsup, Ni ^De, Nch [ABTPSUI] [VDanuT] [RESETAAIDN] 20.3	^Danyb, Nfa Dsup, Ni ^De, Nch [ABTPSUI] [VDanuT] [RESETAAIDN] 20.3.2	^Danyb, Nfa Dsup, Ni ^De, Nch [ABTPSUI] [VDanuT] 20.3.3	^Danyb Nfa, Dch [ABTPSUI] [RBNEXTSB] 21.1	^Danyb Nfa [ABTPSUI] 21.2	^Danyb Nfa, Dch D2pc [ABTPSUI] [RBNEXTSB] 21.3	^Danyb Nfa [ABTPSUI] 21.4
	^Danyb, Nfa Dsup, Ni De, Nch [ABTPSUI] 20.3	^Danyb, Nfa Dsup, Ni De, Nch [ABTPSUI] 20.3.2	^Danyb, Nfa Dsup, Ni De, Nch [ABTPSUI] 20.3.3	^Danyb Nfa, ^Dch [ABTPSUI] 21.1		^Danyb Nfa, Dch Nr, ^D2pc, Dcoor [ABTPSUI] [RBNEXTSB] 21.3	
	^Danyb, Nfa Dsup, Ni ^Nch [ABTPSUI] 20.3	^Danyb, Nfa Dsup, Ni ^Nch [ABTPSUI] 20.3.2	^Danyb, Nfa Dsup, Ni ^Nch [ABTPSUI] 20.3.3			^Danyb Nfa, Dch ^D2pc, ^Dcoor ^Droi, ^Deei [NOTCHAIN] [AABrqUd] [ABDET] [SDETrqF] 21.3	
(Continued on sheet 2 of 87)	^Danyb, Nfa ^Dsup [ABTPSUI] [NOTCHAIN] 20.3	^Danyb, Nfa ^Dsup [ABTPSUI] [NOTCHAIN] 20.3.2	^Danyb, Nfa ^Dsup [ABTPSUI] [NOTCHAIN] 20.3.3			^Danyb Nfa, ^Dch ^Dcoor [AABrqUd] [ABDET] [SDETrqF] 21.3	

Table A.15-1/X.862 – Commitment (sheet 2 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
TP-U-ABORT req *(Concluded 2 of 2)*						^Danyb Nfa, ^Dch Dcoor, D2pc, ^Ncc [ABTPSUI] 21.3	
						^Danyb Nfa, ^Dch Dcoor, D2pc, Ncc [AABrqUd] [ABDET] [SDETrqF] 21.3	
						^Danyb Nfa, ^Dch Dcoor, ^D2pc [AABrqUd] [ABDET] [SDETrqF] 21.3	
						^Danyb Nfa, Dch Droi [COMREQ] [SDETrqF] [ABDET] 21.3	
						^Danyb Nfa, Dch Deei [AABrqUd] [SDETrqF] [ABDET] 21.3	

Table A.15-1/X.862 – Commitment (sheet 3 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT (provider, abortRI) ind or A-P-ABORT ind or A-ABORT ind or A-ABORT req or A-RELEASE (result = affirmative) rsp or A-RELEASE (result = affirmative) cnf	Dsup, Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDETR] [NOTCHAIN] [VNfaT] [CAFPLrqSP] [CRNALL] 99	Dsup, Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDET] [NOTCHAIN] [VNfaT] [VNresultND] [RECCOM-OP] [OWEDONECO] [CRNALL] 21.5.4	Dsup, Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDET] [NOTCHAIN] [VNfaT] [VNresultND] [RECCOM-OP] [OWEDONECO] [CRNALL] 21.5.4	^Dcoor, D2pc ^Danyb, ^Dch [SETDIAGTP] [TPABi] [ABDET] [OWEDONE] [CAFPLrqSB] [COUNTCR] 99			
	Dsup,Dcoor, Danyb ^Db [VDanuF] [ABDETR] [NOTCHAIN] [CAFPLrqSP] 99	Dsup, Dcoor, Danyb ^Db [VDanuF] [ABDET] [NOTCHAIN] [VNresultND] [RECCOM-OP] [OWEDONECO] 21.5.4	Dsup, Dcoor, Danyb ^Db [VDanuF] [ABDET] [NOTCHAIN] [VNresultND] [RECCOM-OP] [OWEDONECO] 21.5.4	^Dcoor, D2pc ^Danyb, Dch [SETDIAGTP] [TPABi] [ABDET] [OWEDONE] [RBNEXTSB] [CAFPLrqSB] [COUNTCR] 99	^Dcoor, D2pc [SETDIAGTP] [TPABi] [ABDET] [OWEDONE] [CAFPLrqSB] [COUNTCR] 99		
	Dsup, Dcoor, Dchat [VDchatF] [NOTCHAIN] [CAFPLrqSP] 99			^Dcoor, D2pc Danyb ^Db [ABDET] [CAFPLrqSB] [COUNTCR] 99			
(Continued on sheet 4 of 87)	Dsup, ^Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDETR] [NOTCHAIN] [RESETAAIDN] [VNfaT] [CRNALL] 20.3	Dsup, ^Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDETR] [NOTCHAIN] [RESETAAIDN] [VNfaT] [CRNALL] 20.3.2		^Dcoor, D2pc Dchat [VDchatF] [CAFPLrqSB] 99			

Table A.15-1/X.862 – Commitment (sheet 4 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT (provider, abortRI) ind or A-P-ABORT ind or A-ABORT ind or A-ABORT req or A-RELEASE (result = affirmative) rsp or A-RELEASE (result = affirmative) cnf *(Continued 2 of 3)*	Dsup,^Dcoor, Danyb ^Db [VDanuF] [ABDETR] [NOTCHAIN] 20.3	Dsup, ^Dcoor, Danyb ^Db [VDanuF] [ABDETR] [NOTCHAIN] 20.3.2		Dcoor, D2pc ^Danyb [SETDIAGTP] [TPABi] [ABDET] [VNfaT] [OWEDONE] [COUNTCR] 21.1		Dcoor, D2pc ^Danyb [SETDIAGTP] [TPABi] [ABDET] [VNfaT] [OWEDONE] 21.1	
	Dsup, ^Dcoor, Dchat [VDchatF] 20.3			Dcoor, D2pc Danyb, ^Db [ABDET] [COUNTCR] 21.1		Dcoor, D2pc Danyb, ^Db [ABDET] 21.1	
	^Dsup,^Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDETR] [NOTCHAIN] [VNfaT] 20.3	^Dsup, ^Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDETR] [NOTCHAIN] [VNfaT] 20.3.2	^Dsup, ^Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDETR] [NOTCHAIN] [VNfaT] 20.3.3	Dcoor, D2pc Db, Dchat [VDchatF] 21.1		Dcoor, D2pc Db, Dchat [VDchatF] 21.1	
	^Dsup, ^Dcoor, Danyb ^Db [ABDETR] [NOTCHAIN] 20.3	^Dsup, ^Dcoor, Danyb ^Db [ABDETR] [NOTCHAIN] 20.3.2	^Dsup, ^Dcoor, Danyb ^Db [ABDETR] [NOTCHAIN] 20.3.3	Dcoor, ^D2pc ^Danyb, ^Dch [SETDIAGTP] [TPABi] [ABDET] [OWEDONE] [COUNTCR] [COUNTCOM] 21.3			
(Continued on sheet 5 of 87)	^Dsup, ^Dcoor, Dchat [VDchatF] 20.3						

Table A.15-1/X.862 – Commitment (sheet 5 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT (provider, abortRI) ind or A-P-ABORT ind or A-ABORT ind or A-ABORT req or A-RELEASE (result = affirmative) rsp or A-RELEASE (result = affirmative) cnf *(Concluded 3 of 3)*	^Dsup, Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDETR] [VNfaT] [CAFPLrqSB] 99	^Dsup, Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDET] [NOTCHAIN] [VNfaT] [VNresultND] [RECCOM-OP] [OWEDONECO] 21.3		Nr, Dcoor, ^D2pc ^Danyb, Dch [SETDIAGTP] [TPABi] [ABDET] [OWEDONE] [RBNEXTSB] [COUNTCR] [COUNTCOM] 21.3			
	^Dsup, Dcoor, Danyb ^Db [ABDETR] [CAFPLrqSB] 99	^Dsup, Dcoor, Danyb ^Db [ABDET] [NOTCHAIN] [VNresultND] [RECCOM-OP] [OWEDONECO] 21.3		Dcoor, ^D2pc Danyb, ^Db,^Dch [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			
	^Dsup, Dcoor, Dchat [VDchatF] [CAFPLrqSB] 99			Nr, Dcoor, ^D2pc Danyb, ^Db, Dch [ABDET] [RBNEXTSB] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			

Table A.15-1/X.862 – Commitment (sheet 6 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
Protocol error or Internal error	Dsup, Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDETR] [NOTCHAIN] [SETDIAG] [AABrqPa] [VNfaT] [CAFPLrqSP] [CRNALL] 99	Dsup, Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDET] [NOTCHAIN] [SETDIAG] [AABrqPa] [VNfaT] [VNresultND] [RECCOM-OP] [OWEDONECO] [CRNALL] 21.5.4	Dsup, Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDET] [NOTCHAIN] [SETDIAG] [AABrqPa] [VNfaT] [VNresultND] [RECCOM-OP] [OWEDONECO] [CRNALL] 21.5.4	^Dcoor, D2pc ^Danyb, ^Dch [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [OWEDONE] [CAFPLrqSB] [COUNTCR] 99	^Dcoor, D2pc [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [OWEDONE] [CAFPLrqSB] [COUNTCR] 99	^Danyb, ^Dch [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [OWEDONE] 21.3	[SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [OWEDONE] 21.3
	Dsup, Dcoor, Danyb ^Db [VDanuF] [ABDETR] [NOTCHAIN] [SETDIAG] [AABrqPa] [CAFPLrqSP] 99	Dsup, Dcoor, Danyb ^Db [VDanuF] [ABDET] [NOTCHAIN] [SETDIAG] [AABrqPa] [VNresultND] [RECCOM-OP] [OWEDONECO] 21.5.4	Dsup, Dcoor, Danyb ^Db [VDanuF] [ABDET] [NOTCHAIN] [SETDIAG] [AABrqPa] [VNresultND] [RECCOM-OP] [OWEDONECO] 21.5.4	^Dcoor, D2pc ^Danyb, Dch [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [OWEDONE] [RBNEXTSB] [CAFPLrqSB] [COUNTCR] 99		D2pc ^Danyb, Dch [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [OWEDONE] [RBNEXTSB] 21.3	
	Dsup, Dcoor Dchat [NOTCHAIN] [SETDIAG] [AABrqPa] [VDchatF] [CAFPLrqSP] 99			^Dcoor, D2pc Danyb ^Db [ABDET] [SETDIAG] [AABrqPa] [CAFPLrqSB] [COUNTCR] 99		Danyb ^Db [ABDET] [SETDIAG] [AABrqPa] 21.3	

(Continued on sheet 7 of 87)

Table A.15-1/X.862 – Commitment (sheet 7 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
Protocol error or Internal error *(Continued 2 of 4)*	Dsup, ^Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDETR] [NOTCHAIN] [RESETAAIDN] [SETDIAG] [AABrqPa] [VNfaT] [CRNALL] 20.3	Dsup, ^Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDETR] [NOTCHAIN] [RESETAAIDN] [SETDIAG] [AABrqPa] [VNfaT] [CRNALL] 20.3.2		^Dcoor, D2pc Dchat [SETDIAG] [AABrqPa] [VDchatF] [CAFPLrqSB] 99		Db 21.3	
	Dsup, ^Dcoor, Danyb ^Db [VDanuF] [ABDETR] [NOTCHAIN] [SETDIAG] [AABrqPa] 20.3	Dsup, ^Dcoor, Danyb ^Db [VDanuF] [ABDETR] [NOTCHAIN] [SETDIAG] [AABrqPa] 20.3.2		Dcoor, D2pc ^Danyb [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [OWEDONE] [COUNTCR] 21.1			
	Dsup, ^Dcoor Dchat [NOTCHAIN] [SETDIAG] [AABrqPa] [VDchatF] 20.3			Dcoor, D2pc Danyb, ^Db [ABDET] [SETDIAG] [AABrqPa] [COUNTCR] 21.1			
(Continued on sheet 8 of 87)				Dcoor, D2pc Db, ^Dchat 21.1			

Table A.15-1/X.862 – Commitment (sheet 8 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
Protocol error or Internal error *(Concluded 3 of 4)*	^Dsup, ^Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDETR] [NOTCHAIN] [SETDIAG] [AABrqPa] [VNfaT] 20.3	^Dsup, ^Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDETR] [NOTCHAIN] [SETDIAG] [AABrqPa] [VNfaT] 20.3.2	^Dsup, ^Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDETR] [NOTCHAIN] [SETDIAG] [AABrqPa] [VNfaT] 20.3.3	Dcoor, D2pc Db, Dchat [VDchatF] 21.1			
	^Dsup, ^Dcoor, Danyb ^Db [ABDETR] [NOTCHAIN] [SETDIAG] [AABrqPa] 20.3	^Dsup, ^Dcoor, Danyb ^Db [ABDETR] [NOTCHAIN] [SETDIAG] [AABrqPa] 20.3.2	^Dsup, ^Dcoor, Danyb ^Db [ABDETR] [NOTCHAIN] [SETDIAG] [AABrqPa] 20.3.3	Dcoor, ^D2pc ^Danyb, ^Dch [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			
	^Dsup, ^Dcoor Db, ^Dchat [NOTCHAIN] 20.3	^Dsup, ^Dcoor Db [NOTCHAIN] 20.3.2	^Dsup, ^Dcoor Db [NOTCHAIN] 20.3.3	Nr, Dcoor, ^D2pc ^Danyb, Dch [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [RBNEXTSB] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		Nr, Dcoor, ^D2pc ^Danyb, Dch [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [RBNEXTSB] [OWEDONE] 21.3	
	^Dsup, ^Dcoor Db, Dchat [NOTCHAIN] [SETDIAG] [AABrqPa] [VDchatF] 20.3						
(Continued on sheet 9 of 87)	^Dsup, Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDETR] [NOTCHAIN] [SETDIAG] [AABrqPa] [VNfaT] [CAFPLrqSB] 99	^Dsup, Dcoor, ^Danyb [SETDIAGTP] [TPABi] [ABDET] [NOTCHAIN] [SETDIAG] [AABrqPa] [VNfaT] [VNresultND] [RECCOM-OP] [OWEDONECO] 21.3				^Dcoor, ^D2pc ^Danyb, Dch [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [OWEDONE] 21.3	

Table A.15-1/X.862 – Commitment (sheet 9 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
Protocol error or Internal error *(Concluded 4 of 4)*	^Dsup, Dcoor, Danyb ^Db [ABDETR] [NOTCHAIN] [SETDIAG] [AABrqPa] [CAFPLrqSB] 99	^Dsup, Dcoor, Danyb ^Db [ABDET] [NOTCHAIN] [SETDIAG] [AABrqPa] [VNresultND] [RECCOM-OP] [OWEDONECO] 21.3		Dcoor, ^D2pc Danyb, ^Db, ^Dch [ABDET] [SETDIAG] [AABrqPa] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		Nr, Dcoor, ^D2pc Danyb, Dch ^Db [ABDET] [SETDIAG] [AABrqPa] 21.3	
	^Dsup, Dcoor Db, ^Dchat [NOTCHAIN] [CAFPLrqSB] 99			Nr, Dcoor, ^D2pc Danyb, Dch [SETDIAGTP] [ABDET] [SETDIAG] [AABrqPa] [RBNEXTSB] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		^Dcoor, ^D2pc Danyb, Dch ^Db [ABDET] [SETDIAG] [AABrqPa] 21.3	
	^Dsup, Dcoor Db, Dchat [NOTCHAIN] [SETDIAG] [AABrqPa] [VDchatF] [CAFPLrqSB] 99						
U-ASE ind	Dsup, Dcoor, ^Danyb [TPABiUP] [ABDETR] [NOTCHAIN] [AABrqPaUP] [VNfaT] [CAFPLrqSP] [CRNALL] 99	Dsup, Dcoor, ^Danyb [TPABiUP] [ABDET] [NOTCHAIN] [AABrqPaUP] [VNfaT] [VNresultND] [RECCOM-OP] [OWEDONECO] [CRNALL] 21.5.4	Dsup, Dcoor, ^Danyb [TPABiUP] [ABDET] [NOTCHAIN] [AABrqPaUP] [VNfaT] [VNresultND] [RECCOM-OP] [OWEDONECO] [CRNALL] 21.5.4				

(Continued on sheet 10 of 87)

Table A.15-1/X.862 – Commitment (sheet 10 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
U-ASE ind *(Concluded 2 of 2)*	Dsup, Dcoor, Danyb ^Db [VDanuF] [ABDETR] [NOTCHAIN] [AABrqPaUP] [CAFPLrqSP] 99	Dsup, Dcoor, Danyb ^Db [VDanuF] [ABDET] [NOTCHAIN] [AABrqPaUP] [VNresultND] [RECCOM-OP] [OWEDONECO] 21.5.4	Dsup, Dcoor, Danyb ^Db [VDanuF] [ABDET] [NOTCHAIN] [AABrqPaUP] [VNresultND] [RECCOM-OP] [OWEDONECO] 21.5.4				
	^Dsup, Dcoor, ^Danyb [TPABiUP] [ABDETR] [NOTCHAIN] [AABrqPaUP] [VNfaT] [CAFPLrqSB] 99	^Dsup, Dcoor, ^Danyb [TPABiUP] [ABDET] [NOTCHAIN] [AABrqPaUP] [VNfaT] [VNresultND] [RECCOM-OP] [OWEDONECO] 21.3					
	^Dsup, Dcoor, Danyb ^Db [ABDETR] [NOTCHAIN] [AABrqPaUP] [CAFPLrqSB] 99	^Dsup, Dcoor, Danyb ^Db [ABDET] [NOTCHAIN] [AABrqPaUP] [VNresultND] [RECCOM-OP] [OWEDONECO] 21.3					
AF-U-ERROR ind or AF-HANDSHAKE ind or AF-REQUEST-CONTROL ind	20.3	20.3.2	20.3.3				

Table A.15-1/X.862 – Commitment (sheet 11 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-DEFER (end-dialogue) ind	Dsup, Dcoor, ^Danyb [TPABiUP] [ABDETR] [NOTCHAIN] [AABrqPaUP] [VNfaT] [CAFPLrqSP] [CRNALL] 99	Dsup, Dcoor, ^Danyb [TPABiUP] [ABDET] [NOTCHAIN] [AABrqPaUP] [VNfaT] [VNresultND] [RECCOM-OP] [OWEDONECO] [CRNALL] 21.5.4	Dsup, Dcoor, ^Danyb [TPABiUP] [ABDET] [NOTCHAIN] [AABrqPaUP] [VNfaT] [VNresultND] [RECCOM-OP] [OWEDONECO] [CRNALL] 21.5.4				
	Dsup, Dcoor, Danyb ^Db [VDanuF] [ABDETR] [NOTCHAIN] [AABrqPaUP] [CAFPLrqSP] 99	Dsup, Dcoor, Danyb ^Db [VDanuF] [ABDET] [NOTCHAIN] [AABrqPaUP] [VNresultND] [RECCOM-OP] [OWEDONECO] 21.5.4	Dsup, Dcoor, Danyb ^Db [VDanuF] [ABDET] [NOTCHAIN] [AABrqPaUP] [VNresultND] [RECCOM-OP] [OWEDONECO] 21.5.4				
AF-PREPARE ind or AF-PREPARE (data-permitted = FALSE) ind or AF-PREPARE (data-permitted = TRUE) ind	Dcoor 20.3	Dcoor 20.3.2	Dcoor, Dsup 20.3.3				
C-READY ind	Ptok, Dcoor [VDcoorF] [CRDYSET] [VNcntrdyDEC] [VNrdyiINC] [VDrdyiT] [DECISION] 20.3	Dcoor 20.3.2	Dcoor 20.3.3				
	^Ptok, Dcoor 20.3						

Table A.15-1/X.862 – Commitment (sheet 12 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
C-NOCHANGE (result-requested) ind	Dcoor [VDcoorF] [VD2pcF] [CRDYRESET] [VNcntrdyDEC] [VNopiINC] [VDopiT] [DECISION] 20.3						
	Dcoor, ^Dsup [VDcoorF] [CRDYRESET] [VD2pcF] [VNcntrdyDEC] [VDroiT] [DECISION] 20.3						
AF-EARLY-EXIT ind		^Dsup, Dcoor, ^Du, ^Danyb [SETDIAGEC] [TRBi] [CRBRS] [INITRB] [OWEDONE] [COUNTRB] 23.2					
	^Dsup, Dcoor, Du, ^Danyb [SETDIAGEC] [TRBi] [CRBRS] [LOGDAMRB] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	^Dsup, Dcoor, Du, ^Danyb [SETDIAGEC] [TRBi] [CRBRS] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2					

(Continued on sheet 13 of 87)

Table A.15-1/X.862 – Commitment (sheet 13 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-EARLY-EXIT ind *(Concluded 2 of 2)*	^Dsup, Dcoor, Dtb [SETDIAGEC] [TRBi] [RBRSPAB] [LOGDAMRB] [INITRB] [OWEDONE] [COUNTRB] [SDETrqF] [ABDET] 23.2	^Dsup, Dcoor, Dtb [SETDIAGEC] [TRBi] [RBRSPAB] [INITRB] [OWEDONE] [COUNTRB] [SDETrqF] [ABDET] 23.2					
C-COMMIT ind	Dsup, Dcoor ^Nch [RECCOM] [OWEDONECO] 21.5						
	Dsup Nch, De [NOTCHAIN] [RECCOM] [OWEDONECO] 21.5						
	^Dsup, Dcoor, Dhrsfu, ^Dcdfu [RECCOM] [OWEDONECO] 21.3						
	^Dsup, Dcoor, Dhrsfu, Dcdfu [RECCOM] [OWEDONECO] 21.1						
	^Dsup, Dcoor, ^Dhrsfu [RECCOM] [OWEDONECO] 21.1						

Table A.15-1/X.862 – Commitment (sheet 14 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
C-NOCHANGE (commit) cnf		Dsup, Dcoor, ^Nch, ^Danyb [VNresultC] [RECCOM-OP] [OWEDONECO] [CPSAP] 21.5.4	Dsup, Dcoor, ^Nch, ^Danyb [VNresultND] [RECCOM-OP] [OWEDONECO] [CPSAP] 21.5.4				
		Dsup, Dcoor, ^Nch, Dtb [VNresultC] [RECCOM-OP] [AABrqUd] [SDETrqF] [ABDET] [OWEDONECO] 21.5.4	Dsup, Dcoor, ^Nch, Dtb [VNresultND] [RECCOM-OP] [AABrqUd] [SDETrqF] [ABDET] [OWEDONECO] 21.5.4				
		^Dsup, Dcoor, ^Dch, ^Danyb Dhrsfu, ^Dcdfu [VNresultC] [RECCOM-OP] [OWEDONECO] 21.3	Dsup, Dcoor, Nch, De [VNresultND] [NOTCHAIN] [RECCOM-OP] [SDETrqF] [ABDET] [OWEDONECO] 21.5.4				
		^Dsup, Dcoor, ^Dch, ^Danyb Dhrsfu, Dcdfu [VNresultC] [RECCOM-OP] [OWEDONECO] 21.1	Dsup, Dcoor, Nch, ^De 20.3.3				
		^Dsup, Dcoor, ^Dch, ^Danyb ^Dhrsfu [VNresultC] [RECCOM-OP] [OWEDONECO] 21.1					

(Continued on sheet 15 of 87)

Table A.15-1/X.862 – Commitment (sheet 15 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
C-NOCHANGE (commit) cnf *(Continued 2 of 4)*		^Dsup, Dcoor, ^Dch, Dtb Dhrsfu, ^Dcdfu [VNresultC] [RECCOM-OP] [AABrqUd] [SDETrqF] [ABDET] [OWEDONECO] 21.3					
		^Dsup, Dcoor, ^Dch, Dtb Dhrsfu, Dcdfu [VNresultC] [RECCOM-OP] [OWEDONECO] 21.1					
		^Dsup, Dcoor, ^Dch, Dtb ^Dhrsfu [VNresultC] [RECCOM-OP] [OWEDONECO] 21.1					
(Continued on sheet 16 of 87)		^Dsup, Dcoor, Dch, ^De, ^Danyb Dhrsfu, ^Dcdfu [VNresultC] [RECCOM-OP] [OWEDONECO] 21.3					

Table A.15-1/X.862 – Commitment (sheet 16 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
C-NOCHANGE (commit) cnf *(Continued 3 of 4)*		^Dsup, Dcoor, Dch, ^De, ^Danyb Dhrsfu, Dcdfu [VNresultC] [RECCOM-OP] [OWEDONECO] 21.1					
		^Dsup, Dcoor, Dch, ^De, ^Danyb ^Dhrsfu [VNresultC] [RECCOM-OP] [OWEDONECO] 21.1					
		^Dsup, Dcoor, Dch, ^De, Dtb Dhrsfu, ^Dcdfu [VNresultC] [RECCOM-OP] [AABrqUd] [SDETrqF] [ABDET] [OWEDONECO] 21.3					
		^Dsup, Dcoor, Dch, ^De, Dtb Dhrsfu, Dcdfu [VNresultC] [RECCOM-OP] [OWEDONECO] 21.1					
(Continued on sheet 17 of 87)		^Dsup, Dcoor, Dch, ^De, Dtb ^Dhrsfu [VNresultC] [RECCOM-OP] [OWEDONECO] 21.1					

Table A.15-1/X.862 – Commitment (sheet 17 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
C-NOCHANGE (commit) cnf *(Concluded 4 of 4)*		^Dsup, Dcoor, Dch, De Dhrsfu, ^Dcdfu [NOTCHAIN] [VNresultC] [RECCOM-OP] [OWEDONECO] [SDETrqF] [ABDET] 21.3					
		^Dsup, Dcoor, Dch, De Dhrsfu, Dcdfu [NOTCHAIN] [VNresultC] [RECCOM-OP] [OWEDONECO] 21.1					
		^Dsup, Dcoor, Dch, De ^Dhrsfu [NOTCHAIN] [VNresultC] [RECCOM-OP] [OWEDONECO] 21.1					
C-NOCHANGE (no-change) cnf		Dsup, Dcoor, ^Nch, ^Danyb [VNresultNC] [RECCOM-OP] [OWEDONECO] [CPSAP] 21.5.4	Dsup, Dcoor, ^Nch, ^Danyb [VNresultND] [RECCOM-OP] [OWEDONECO] [CPSAP] 21.5.4				
		Dsup, Dcoor, ^Nch, Dtb [VNresultNC] [RECCOM-OP] [AABrqUd] [SDETrqF] [ABDET] [OWEDONECO] 21.5.4	Dsup, Dcoor, ^Nch, Dtb [VNresultND] [RECCOM-OP] [AABrqUd] [SDETrqF] [ABDET] [OWEDONECO] 21.5.4				

(Continued on sheet 18 of 87)

Table A.15-1/X.862 – Commitment (sheet 18 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
C-NOCHANGE (no-change) cnf *(Concluded 2 of 2)*		^Dsup, Dcoor, ^Dch, ^Danyb [VNresultNC] [RECCOM-OP] [OWEDONECO] 21.3	Dsup, Dcoor, Nch, De [VNresultND] [NOTCHAIN] [RECCOM-OP] [SDETrqF] [ABDET] [OWEDONECO] 21.5.4				
		^Dsup, Dcoor, ^Dch, Dtb [VNresultNC] [RECCOM-OP] [AABrqUd] [SDETrqF] [ABDET] [OWEDONECO] 21.3	Dsup, Dcoor, Nch, ^De 20.3.3				
		^Dsup, Dcoor, Dch, ^De, ^Danyb [VNresultNC] [RECCOM-OP] [OWEDONECO] 21.3					
		^Dsup, Dcoor, Dch, ^De, Dtb [VNresultNC] [RECCOM-OP] [OWEDONECO] [SDETrqF] [ABDET] 21.3					
		^Dsup, Dcoor, Dch, De [NOTCHAIN] [VNresultNC] [RECCOM-OP] [OWEDONECO] [SDETrqF] [ABDET] 21.3					

Table A.15-1/X.862 – Commitment (sheet 19 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
C-NOCHANGE (not-determined) cnf		Dsup, Dcoor, ^Nch, ^Danyb [VNresultND] [RECCOM-OP] [OWEDONECO] [CPSAP] 21.5.4	Dsup, Dcoor, ^Nch, ^Danyb [VNresultND] [RECCOM-OP] [OWEDONECO] [CPSAP] 21.5.4				
		Dsup, Dcoor, ^Nch, Dtb [VNresultND] [RECCOM-OP] [AABrqUd] [SDETrqF] [ABDET] [OWEDONECO] 21.5.4	Dsup, Dcoor, ^Nch, Dtb [VNresultND] [RECCOM-OP] [AABrqUd] [SDETrqF] [ABDET] [OWEDONECO] 21.5.4				
		^Dsup, Dcoor, ^Dch, ^Danyb [VNresultND] [RECCOM-OP] [OWEDONECO] 21.3	Dsup, Dcoor, Nch, De [VNresultND] [NOTCHAIN] [RECCOM-OP] [SDETrqF] [ABDET] [OWEDONECO] 21.5.4				
		^Dsup, Dcoor, ^Dch, Dtb [VNresultND] [RECCOM-OP] [AABrqUd] [SDETrqF] [ABDET] [OWEDONECO] 21.3					

(Continued on sheet 20 of 87)

Table A.15-1/X.862 – Commitment (sheet 20 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
C-NOCHANGE (not-determined) cnf *(Concluded 2 of 2)*		^Dsup, Dcoor, Dch, ^De, ^Danyb [VNresultND] [RECCOM-OP] [OWEDONECO] 21.3					
		^Dsup, Dcoor, Dch, ^De, Dtb [VNresultND] [RECCOM-OP] [OWEDONECO] 21.3					
		^Dsup, Dcoor, Dch, De [NOTCHAIN] [VNresultND] [RECCOM-OP] [OWEDONECO] [SDETrqF] [ABDET] 21.3					
C-BEGIN ind			Dsup, Dcoor, ^Danyb [VNresultND] [RECCOM-OP] [OWEDONECO] [CBEAFTCO] [CPSAP] 21.5.4				
			Dsup, Dcoor, Danu [VNresultND] [RECCOM-OP] [OWEDONECO] [SDETrqRBC] [ABDET] 21.5.4				
C-BEGIN cnf	^Dsup, Dcoor [VDbcrT] [VDxF] 20.3	^Dsup, Dcoor [VDbcrT] [VDxF] 20.3.2					

Table A.15-1/X.862 – Commitment (sheet 21 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
C-COMMIT+C-BEGIN ind	Dsup, Dcoor Nch, ^De [RECCOM] [OWEDONECO] 21.5						
AF-ABORT (user, commitRI) ind	Dsup, Dcoor, ^Dtb [NOTCHAIN] [RECCOM] [TUABi] [ABPTNR] [OWEDONECO] 21.5						
	Dsup, Dcoor, Dtb [NOTCHAIN] [RECCOM] [ABPTNR] [OWEDONECO] [VDanuF] 21.5						
	^Dsup, Dcoor, ^Dtb Dhrsfu, ^Dcdfu [NOTCHAIN] [RECCOM] [TUABi] [ABPTNR] [OWEDONECO] 21.3						
	^Dsup, Dcoor, Dtb Dhrsfu, ^Dcdfu [NOTCHAIN] [RECCOM] [ABPTNR] [OWEDONECO] 21.3						

Table A.15-1/X.862 – Commitment (sheet 22 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT (nochangeRC, commit) ind		Dsup, Dcoor, ^Nch, ^Danyb [VNresultC] [RECCOM-OP] [TUABi] [OWEDONECO] [SDETrqF] [ABDET] 21.5.4					
		Dsup, Dcoor, ^Nch, Dtb [VNresultC] [RECCOM-OP] [AABrqUd] [SDETrqF] [ABDET] [OWEDONECO] 21.5.4					
		^Dsup, Dcoor, ^Dch, ^Danyb Dhrsfu, ^Dcdfu [VNresultC] [RECCOM-OP] [TUABi] [OWEDONECO] [SDETrqF] [ABDET] 21.3					
		^Dsup, Dcoor, ^Dch, Dtb Dhrsfu, ^Dcdfu [VNresultC] [RECCOM-OP] [ABPTNR] [OWEDONECO] [SDETrqF] [ABDET] 21.3					

(Continued on sheet 23 of 87)

Table A.15-1/X.862 – Commitment (sheet 23 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT (nochangeRC, commit) ind *(Concluded 2 of 2)*		Nr, ^Dsup, Dcoor, Dch, ^Danyb Dhrsfu, ^Dcdfu [NOTCHAIN] [VNresultC] [RECCOM-OP] [OWEDONECO] [SDETrqF] [ABDET] 21.3					
		Nr, ^Dsup, Dcoor, Dch, Dtb Dhrsfu, ^Dcdfu [NOTCHAIN] [VNresultC] [RECCOM-OP] [OWEDONECO] [SDETrqF] [ABDET] 21.3					
AF-ABORT (nochangeRC, not-determined) ind		Dsup, Dcoor, ^Nch, ^Danyb [VNresultND] [RECCOM-OP] [TUABi] [OWEDONECO] [SDETrqF] [ABDET] 21.5.4	Dsup, Dcoor, ^Danyb [NOTCHAIN] [VNresultND] [RECCOM-OP] [TUABi] [OWEDONECO] [SDETrqF] [ABDET] 21.5.4				
		Dsup, Dcoor, ^Nch, Dtb [VNresultND] [RECCOM-OP] [OWEDONECO] [SDETrqF] [ABDET] 21.5.4	Dsup, Dcoor, ^Nch, Dtb [VNresultND] [RECCOM-OP] [OWEDONECO] [SDETrqF] [ABDET] 21.5.4				

(Continued on sheet 24 of 87)

Table A.15-1/X.862 – Commitment (sheet 24 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT (nochangeRC, not-determined) ind *(Concluded 2 of 2)*		^Dsup, ^Dch, Dcoor, ^Danyb [VNresultND] [RECCOM-OP] [TUABi] [OWEDONECO] [SDETrqF] [ABDET] 21.3	Dsup, Nch, Dcoor, Danu [VDanuF] [NOTCHAIN] [VNresultND] [RECCOM-OP] [OWEDONECO] [SDETrqF] [ABDET] 21.5.4				
		^Dsup, ^Dch, Dcoor, Dtb [VNresultND] [RECCOM-OP] [ABPTNR] [OWEDONECO] [SDETrqF] [ABDET] 21.3					
		Nr, ^Dsup, Dch, Dcoor, ^Danyb [NOTCHAIN] [VNresultND] [RECCOM-OP] [OWEDONECO] [SDETrqF] [ABDET] 21.3					
		Nr, ^Dsup, Dch, Dcoor, Dtb [NOTCHAIN] [VNresultND] [RECCOM-OP] [OWEDONECO] [SDETrqF] [ABDET] 21.3					

Table A.15-1/X.862 – Commitment (sheet 25 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-EARLY-EXIT cnf			Dsup, Dcoor ^Dch, ^Dtb [VDanuF] [VNresultND] [RECCOM-OP] [OWEDONECO] [CPSAP] 21.5.4				
			Dsup, Dcoor ^Dch, Dtb [VDanuF] [VNresultND] [RECCOM-OP] [ABDET] [AABrqUd] [SDETrqBF] [OWEDONECO] 21.5.4				
			Dsup, Dcoor Dch, ^Dtb [VDanuF] [VNresultND] [RECCOM-OP] [VNcntINC] [OWEDONECO] 21.5.3				
			Dsup, Dcoor Dch, Dtb [VDanuF] [NOTCHAIN] [VNresultND] [RECCOM-OP] [ABDET] [AABrqUd] [SDETrqBF] [OWEDONECO] 21.5.4				

Table A.15-1/X.862 – Commitment (sheet 26 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
C-ROLLBACK cnf			Dsup, Dcoor ^Dch [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] [CPSAP] 23.7				
			Dsup, Dcoor ^Dch, Dtb [SETDIAGSP] [TRBi] [INITRB] [ABDET] [AABrqUd] [SDETrqBF] [OWEDONE] 23.7				
			Dsup, Dcoor Dch, ^Dtb [VDanuF] [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.6				
			Dsup, Dcoor Dch, Dtb [VDanuF] [SETDIAGSP] [TRBi] [INITRB] [ABDET] [AABrqUd] [SDETrqBF] [OWEDONE] 23.6				

Table A.15-1/X.862 – Commitment (sheet 27 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT (user, rollbackRC) ind			Dcoor, Dsup ^Danyb [VDanuF] [TUABiR] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.7				
			Dcoor, Dsup Danyb [VDanuF] [SETDIAGSP] [TRBi] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] 23.7				
TP-DONE (heuristic-report) req				Dd, ^Dfdone Ncmtr [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.1	Dd, ^Dfdone Ncmtr [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.2	Dd, ^Dfdone Ncmtr [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.3	Dd, ^Dfdone Ncmtr [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.4
TP-DONE (heuristic-report, completion-report) req				Dd, ^Dfdone Ncmtr, Dcrpa [LOGDAM] [SAVECR] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.1	Dd, ^Dfdone Ncmtr, Dcrpa [LOGDAM] [SAVECR] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.2	Dd, ^Dfdone Ncmtr, Dcrpa [LOGDAM] [SAVECR] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.3	Dd, ^Dfdone Ncmtr, Dcrpa [LOGDAM] [SAVECR] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.4

Table A.15-1/X.862 – Commitment (sheet 28 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
TP-DONE (completion-report) req				Dd Dcrpa [SAVECR] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.1	Dd Dcrpa [SAVECR] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.2	Dd Dcrpa [SAVECR] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.3	Dd Dcrpa [SAVECR] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.4
TP-DONE req				Dd [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.1	Dd [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.2	Dd [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.3	Dd [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.4
C-COMMIT cnf				^Dcoor, D2pc ^Danyb, ^De [COUNTREP] [COUNTCOM] [CPSAP] 21.3	^Dcoor, D2pc ^Danyb [CRBrq] [COUNTREP] [COUNTCOM] [CPSAP] 21.4		
				^Dcoor, D2pc ^Danyb, De [SDETrqF] [ABDET] [COUNTREP] [COUNTCOM] 21.3			
				^Dcoor, D2pc ^Dbpart, Dtb [AABrqUd] [SDETrqF] [ABDET] [COUNTREP] [COUNTCOM] 21.3	^Dcoor, D2pc ^Dbpart, Dtb [AABrqUr] [SDETrqRBC] [ABDET] [COUNTREP] [COUNTCOM] 21.3		

(Continued on sheet 29 of 87)

Table A.15-1/X.862 – Commitment (sheet 29 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
C-COMMIT cnf *(Concluded 2 of 2)*				^Dcoor, D2pc Dbpart [SDETrqF] [ABDET] [COUNTREP] [COUNTCOM] 21.3			
AF-REPORT (commitRC, heuristic-report) ind				^Dcoor, D2pc ^Danyb, ^De ^Dhrsfu [TREP] [LOGDAM] [COUNTREP] [COUNTCOM] [CPSAP] 21.3	^Dcoor, D2pc ^Danyb ^Dhrsfu [TREP] [LOGDAM] [CRBrq] [COUNTREP] [COUNTCOM] [CPSAP] 21.4		
				^Dcoor, D2pc ^Danyb, De ^Dhrsfu [TREP] [LOGDAM] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			
				^Dcoor, D2pc ^Dbpart, Dtb ^Dhrsfu [TREP] [LOGDAM] [AABrqUd] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3	^Dcoor, D2pc ^Dbpart, Dtb ^Dhrsfu [TREP] [LOGDAM] [AABrqUr] [SDETrqRBC] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		

(Continued on sheet 30 of 87)

Table A.15-1/X.862 – Commitment (sheet 30 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-REPORT (commitRC, heuristic-report) ind *(Concluded 2 of 2)*				^Dcoor, D2pc Dbpart ^Dhrsfu [TREP] [LOGDAM] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			
AF-REPORT (commitRC, heuristic-report, completion-report) ind				^Dcoor, D2pc ^Danyb, ^De ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [COUNTREP] [COUNTCOM] [CPSAP] 21.3	^Dcoor, D2pc ^Danyb ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [CRBrq] [COUNTREP] [COUNTCOM] [CPSAP] 21.4		
(Continued on sheet 31 of 87)				^Dcoor, D2pc ^Danyb, De ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			

Table A.15-1/X.862 – Commitment (sheet 31 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-REPORT (commitRC, heuristic-report, completion-report) ind *(Concluded 2 of 2)*				^Dcoor, D2pc ^Dbpart, Dtb ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [AABrqUd] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3	^Dcoor, D2pc ^Dbpart, Dtb ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [AABrqUr] [SDETrqRBC] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		
				^Dcoor, D2pc Dbpart ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			

Table A.15-1/X.862 – Commitment (sheet 32 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-REPORT (commitRC, completion-report) ind				^Dcoor, D2pc ^Danyb, ^De Dcdfu [TREP] [COUNTREP] [COUNTCOM] [CPSAP] 21.3	^Dcoor, D2pc ^Danyb Dcdfu [TREP] [CRBrq] [COUNTREP] [COUNTCOM] [CPSAP] 21.4		
				^Dcoor, D2pc ^Danyb, De Dcdfu [TREP] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			
				^Dcoor, D2pc ^Dbpart, Dtb Dcdfu [TREP] [AABrqUd] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3	^Dcoor, D2pc ^Dbpart, Dtb Dcdfu [TREP] [AABrqUr] [SDETrqRBC] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		
				^Dcoor, D2pc Dbpart Dcdfu [TREP] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			

Table A.15-1/X.862 – Commitment (sheet 33 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT (user, commitRC) ind				^Dcoor, D2pc ^Danyb, Du [TUABi] [SDETrqF] [ABDET] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			
				^Dcoor, D2pc Danyb, Du ^Dbpart [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			
				^Dcoor, D2pc ^Danyb, Dch [TUABi] [SDETrqRB] [ABDET] [VDanuT] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3	^Dcoor, D2pc ^Danyb [TUABi] [SDETrqRB] [ABDET] [VDanuT] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		
				^Dcoor, D2pc Danyb, ^Du ^Dbpart [SDETrqRB] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3	^Dcoor, D2pc Danyb, ^Dbpart [SDETrqRB] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		

Table A.15-1/X.862 – Commitment (sheet 34 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT-AND-REPORT (commitRC, heuristic-report) ind				^Dcoor, D2pc ^Danyb, Du ^Dhrsfu [TUABi] [TREP] [LOGDAM] [SDETrqF] [ABDET] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			
				^Dcoor, D2pc Danyb, Du ^Dbpart ^Dhrsfu [TREP] [LOGDAM] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			
(Continued on sheet 35 of 87)				^Dcoor, D2pc ^Danyb, Dch ^Dhrsfu [TUABi] [TREP] [LOGDAM] [SDETrqRB] [ABDET] [VDanuT] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3	^Dcoor, D2pc ^Danyb ^Dhrsfu [TUABi] [TREP] [LOGDAM] [SDETrqRB] [ABDET] [VDanuT] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		

Table A.15-1/X.862 – Commitment (sheet 35 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT-AND-REPORT (commitRC, heuristic-report) ind *(Concluded 2 of 2)*				^Dcoor, D2pc Danyb, ^Du ^Dbpart ^Dhrsfu [TREP] [LOGDAM] [SDETrqRB] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3	^Dcoor, D2pc Danyb ^Dbpart ^Dhrsfu [TREP] [LOGDAM] [SDETrqRB] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		
AF-ABORT-AND-REPORT (commitRC, heuristic-report, completion-report) ind				^Dcoor, D2pc ^Danyb, Du ^Dhrsfu, Dcdfu [TUABi] [TREP] [LOGDAM] [SDETrqF] [ABDET] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			
(Continued on sheet 36 of 87)				^Dcoor, D2pc Danyb, Du ^Dbpart ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			

Table A.15-1/X.862 – Commitment (sheet 36 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT-AND-REPORT (commitRC, heuristic-report, completion-report) ind *(Concluded 2 of 2)*				^Dcoor, D2pc ^Danyb, Dch ^Dhrsfu, Dcdfu [TUABi] [TREP] [LOGDAM] [SDETrqRB] [ABDET] [VDanuT] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3	^Dcoor, D2pc ^Danyb ^Dhrsfu, Dcdfu [TUABi] [TREP] [LOGDAM] [SDETrqRB] [ABDET] [VDanuT] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		
				^Dcoor, D2pc Danyb, ^Du ^Dbpart ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [SDETrqRB] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3	^Dcoor, D2pc Danyb ^Dbpart ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [SDETrqRB] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		
AF-ABORT-AND-REPORT (commitRC, completion-report) ind *(Continued on sheet 37 of 87)*				^Dcoor, D2pc ^Danyb, Du Dcdfu [TUABi] [TREP] [SDETrqF] [ABDET] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			

Table A.15-1/X.862 – Commitment (sheet 37 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT-AND-REPORT (commitRC, completion-report) ind *(Concluded 2 of 2)*				^Dcoor, D2pc Danyb, Du ^Dbpart Dcdfu [TREP] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			
				^Dcoor, D2pc ^Danyb, Dch Dcdfu [TUABi] [TREP] [SDETrqRB] [ABDET] [VDanuT] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3	^Dcoor, D2pc ^Danyb Dcdfu [TUABi] [TREP] [SDETrqRB] [ABDET] [VDanuT] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		
				^Dcoor, D2pc Danyb, ^Du ^Dbpart Dcdfu [TREP] [SDETrqRB] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3	^Dcoor, D2pc Danyb ^Dbpart Dcdfu [TREP] [SDETrqRB] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		

Table A.15-1/X.862 – Commitment (sheet 38 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-REPORT (data, heuristic-report) ind				Dcoor, D2pc ^Dchat ^Dhrsfu [TREP] [LOGDAM] [CPSAP] [COUNTREP] [COUNTCOM] [EARLYC] 21.3			
				Dcoor, ^D2pc ^Danyb ^Dhrsfu [TREP] [LOGDAM] [CPSAP] [COUNTREP] [COUNTCOM] 21.3	Nr, Dcoor, ^D2pc ^Danyb ^Dhrsfu [TREP] [LOGDAM] [CPSAP] [COUNTREP] [COUNTCOM] 21.4		
				Dcoor, ^D2pc Danyb ^Dhrsfu [TREP] [LOGDAM] [ABDET] [AABrqUd] [SDETrqF] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			
					Nr, Dcoor, ^D2pc Dtb ^Dhrsfu [TREP] [LOGDAM] [CPSAP] [COUNTREP] [COUNTCOM] 21.4		

Table A.15-1/X.862 – Commitment (sheet 39 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-REPORT (data, heuristic-report, completion-report) ind				Dcoor, D2pc ^Dchat ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [CPSAP] [COUNTREP] [COUNTCOM] [EARLYC] 21.3	Nr, Dcoor, ^D2pc ^Danyb ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [CPSAP] [COUNTREP] [COUNTCOM] 21.4		
				Dcoor, ^D2pc ^Danyb ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [CPSAP] [COUNTREP] [COUNTCOM] 21.3			
				Dcoor, ^D2pc Danyb ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [ABDET] [AABrqUd] [SDETrqF] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			
					Nr, Dcoor, ^D2pc Danyb ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [CPSAP] [COUNTREP] [COUNTCOM] 21.4		

Table A.15-1/X.862 – Commitment (sheet 40 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-REPORT (data, completion-report) ind				Dcoor, D2pc ^Dchat Dcdfu [TREP] [CPSAP] [COUNTREP] [COUNTCOM] [EARLYC] 21.3	Nr, Dcoor, ^D2pc ^Danyb Dcdfu [TREP] [CPSAP] [COUNTREP] [COUNTCOM] 21.4		
				Dcoor, ^D2pc ^Danyb Dcdfu [TREP] [CPSAP] [COUNTREP] [COUNTCOM] 21.3			
				Dcoor, ^D2pc Danyb Dcdfu [TREP] [ABDET] [AABrqUd] [SDETrqF] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3			
					Nr, Dcoor, ^D2pc Danyb Dcdfu [TREP] [CPSAP] [COUNTREP] [COUNTCOM] 21.4		

Table A.15-1/X.862 – Commitment (sheet 41 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT-AND-REPORT (data, heuristic-report) ind				Dcoor, D2pc ^Danyb ^Dhrsfu [TUABi] [TREP] [LOGDAM] [ABPTNR] [VDanuT] [CPSAP] [COUNTREP] [COUNTCOM] [EARLYC] 21.3			
				Dcoor, D2pc Danyb ^Dhrsfu [TREP] [LOGDAM] [ABPTNR] [VDanuT] [CPSAP] [COUNTREP] [COUNTCOM] [EARLYC] 21.3			
				Dcoor, ^D2pc ^Danyb ^Dhrsfu [NOTCHAIN] [TUABi] [TREP] [LOGDAM] [ABPTNR] [VDanuT] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3	Nr, Dcoor, ^D2pc ^Danyb ^Dhrsfu [NOTCHAIN] [TUABi] [TREP] [LOGDAM] [ABPTNR] [VDanuT] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		

(Continued on sheet 42 of 87)

Table A.15-1/X.862 – Commitment (sheet 42 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT-AND-REPORT (data, heuristic-report) ind *(Concluded 2 of 2)*				Dcoor, ^D2pc Danyb ^Dhrsfu [NOTCHAIN] [TREP] [LOGDAM] [ABPTNR] [VDanuT] [SDETrqF] [ABDET] [COUNTREP] [COUNTREP] [COUNTCOM] 21.3	Nr, Dcoor, ^D2pc Danyb ^Dhrsfu [NOTCHAIN] [TREP] [LOGDAM] [ABPTNR] [VDanuT] [SDETrqF] [ABDET] [COUNTREP] [COUNTREP] [COUNTCOM] 21.3		
AF-ABORT-AND-REPORT (data, heuristic-report, completion-report) ind				Dcoor, D2pc ^Danyb ^Dhrsfu, Dcdfu [TUABi] [TREP] [LOGDAM] [ABPTNR] [VDanuT] [CPSAP] [COUNTREP] [COUNTCOM] [EARLYC] 21.3			
(Continued on sheet 43 of 87)				Dcoor, D2pc Danyb ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [ABPTNR] [VDanuT] [CPSAP] [COUNTREP] [COUNTCOM] [EARLYC] 21.3			

Table A.15-1/X.862 – Commitment (sheet 43 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT-AND-REPORT (data, heuristic-report, completion-report) ind *(Concluded 2 of 2)*				Dcoor, ^D2pc ^Danyb ^Dhrsfu, Dcdfu [NOTCHAIN] [TUABi] [TREP] [LOGDAM] [ABPTNR] [VDanuT] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3	Nr, Dcoor, ^D2pc ^Danyb ^Dhrsfu, Dcdfu [NOTCHAIN] [TUABi] [TREP] [LOGDAM] [ABPTNR] [VDanuT] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		
				Dcoor, ^D2pc Danyb ^Dhrsfu, Dcdfu [NOTCHAIN] [TREP] [LOGDAM] [ABPTNR] [VDanuT] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3	Nr, Dcoor, ^D2pc Danyb ^Dhrsfu, Dcdfu [NOTCHAIN] [TREP] [LOGDAM] [ABPTNR] [VDanuT] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		
AF-ABORT-AND-REPORT (data, completion-report) ind *(Continued on sheet 44 of 87)*				Dcoor, D2pc ^Danyb Dcdfu [TUABi] [TREP] [ABPTNR] [VDanuT] [CPSAP] [COUNTREP] [COUNTCOM] [EARLYC] 21.3			

Table A.15-1/X.862 – Commitment (sheet 44 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT-AND-REPORT (data, completion-report) ind *(Concluded 2 of 2)*				Dcoor, D2pc Danyb Dcdfu [TREP] [ABPTNR] [VDanuT] [CPSAP] [COUNTREP] [COUNTCOM] [EARLYC] 21.3			
				Dcoor, ^D2pc ^Danyb Dcdfu [NOTCHAIN] [TREP] [ABPTNR] [VDanuT] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3	Nr, Dcoor, ^D2pc ^Danyb Dcdfu [NOTCHAIN] [TREP] [ABPTNR] [VDanuT] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		
				Dcoor, ^D2pc Danyb Dcdfu [NOTCHAIN] [TREP] [VDanuT] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3	Nr, Dcoor, ^D2pc Danyb Dcdfu [NOTCHAIN] [TREP] [VDanuT] [SDETrqF] [ABDET] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		

Table A.15-1/X.862 – Commitment (sheet 45 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
C-ROLLBACK ind	Dsup, Dcoor [VDanuF] [SETDIAGSP] [TRBi] [LOGDAMRB] [INITRB] [OWEDONE] 23.4	Dsup, Dcoor [VDanuF] [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.4	Dsup, Dcoor [VDanuF] [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.4				
		^Dsup, Dcoor, ^Du, ^Danyb [SETDIAGSB] [TRBi] [CRBRS] [INITRB] [OWEDONE] [COUNTRB] 23.2					
	^Dsup, Dcoor, Du, ^Danyb [SETDIAGSB] [TRBi] [CRBRS] [LOGDAMRB] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	^Dsup, Dcoor, Du, ^Danyb [SETDIAGSB] [TRBi] [CRBRS] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2					
	^Dsup, Dcoor, Dtb [SETDIAGSB] [TRBi] [RBRSPAB] [LOGDAMRB] [INITRB] [OWEDONE] [COUNTRB] [SDETrqF] [ABDET] 23.2	^Dsup, Dcoor, Dtb [SETDIAGSB] [TRBi] [RBRSPAB] [INITRB] [OWEDONE] [COUNTRB] [SDETrqF] [ABDET] 23.2					

Table A.15-1/X.862 – Commitment (sheet 46 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
C-CANCEL ind	Dsup, Dcoor [VDcancrT] [VDanuF] [SETDIAGSP] [TRBi] [LOGDAMRB] [INITRB] [OWEDONE] 23.3	Dsup, Dcoor [VDcancrT] [VDanuF] [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.3	^Neer, Dsup, Dcoor [VDcancrT] [VDanuF] [SETDIAGSP] [TRBi] [INITRB] [OWEDONE] 23.3				
	^Dsup, Dcoor [VDcancrT] [SETDIAGSB] [TRBi] [RBREQ] [LOGDAMRB] [INITRB] [OWEDONE] 23.1	^Dsup, Dcoor [VDcancrT] [SETDIAGSB] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1					
AF-ABORT (user, rollbackRI) ind	^Danyb, Dsup, Dcoor [TUABiR] [LOGDAMRB] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4	^Danyb, Dsup, Dcoor [TUABiR] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4	^Danyb, Dsup, Dcoor [TUABiR] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4				
(Continued on sheet 47 of 87)	Danyb, Dsup, Dcoor [VDanuF] [SETDIAGSP] [TRBi] [LOGDAMRB] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4	Danyb, Dsup, Dcoor [VDanuF] [SETDIAGSP] [TRBi] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4	Danyb, Dsup, Dcoor [VDanuF] [SETDIAGSP] [TRBi] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4				

Table A.15-1/X.862 – Commitment (sheet 47 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT (user, rollbackRI) ind *(Concluded 2 of 2)*	^Dsup, Dcoor, ^Danyb [TUABiR] [CRBRS] [LOGDAMRB] [INITRB] [OWEDONE] [COUNTRB] [SDETrqF] [ABDET] 23.2	^Dsup, Dcoor, ^Danyb [TUABiR] [CRBRS] [INITRB] [OWEDONE] [COUNTRB] [SDETrqF] [ABDET] 23.2					
	^Dsup, Dcoor, Dtb [SETDIAGSB] [TRBi] [CRBRS] [LOGDAMRB] [INITRB] [OWEDONE] [COUNTRB] [SDETrqF] [ABDET] 23.2	^Dsup, Dcoor, Dtb [SETDIAGSB] [TRBi] [CRBRS] [INITRB] [OWEDONE] [COUNTRB] [SDETrqF] [ABDET] 23.2					
AF-ABORT (provider, rollbackRI) ind	^Danyb, Dsup, Dcoor [TPABiR] [LOGDAMRB] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4	^Danyb, Dsup, Dcoor [TPABiR] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4	^Danyb, Dsup, Dcoor [TPABiR] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4				
(Continued on sheet 48 of 87)	Danyb, Dsup, Dcoor [VDanuF] [SETDIAGSP] [TRBi] [LOGDAMRB] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4	Danyb, Dsup, Dcoor [VDanuF] [SETDIAGSP] [TRBi] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4	Danyb, Dsup, Dcoor [VDanuF] [SETDIAGSP] [TRBi] [ABPTNR] [NOTCHAIN] [INITRB] [OWEDONE] 23.4				

Table A.15-1/X.862 – Commitment (sheet 48 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT (provider, rollbackRI) ind *(Concluded 2 of 2)*	^Dsup, Dcoor, ^Danyb [TPABiR] [CRBRS] [LOGDAMRB] [INITRB] [OWEDONE] [COUNTRB] [SDETrqF] [ABDET] 23.2	^Dsup, Dcoor, ^Danyb [TPABiR] [CRBRS] [INITRB] [OWEDONE] [COUNTRB] [SDETrqF] [ABDET] 23.2					
	^Dsup, Dcoor, Dtb [SETDIAGSB] [TRBi] [CRBRS] [LOGDAMRB] [INITRB] [OWEDONE] [COUNTRB] [SDETrqF] [ABDET] 23.2	^Dsup, Dcoor, Dtb [SETDIAGSB] [TRBi] [CRBRS] [INITRB] [OWEDONE] [COUNTRB] [SDETrqF] [ABDET] 23.2					
AF-REPORT (rollbackRI, heuristic-report) ind *(Continued on sheet 49 of 87)*		^Dsup, Dcoor ^Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2					

Table A.15-1/X.862 – Commitment (sheet 49 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-REPORT (rollbackRI, heuristic-report) ind *(Concluded 2 of 2)*	^Dsup, Dcoor Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	^Dsup, Dcoor Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2					
AF-REPORT (rollbackRI, heuristic-report, completion-report) ind		^Dsup, Dcoor ^Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2					
	^Dsup, Dcoor Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	^Dsup, Dcoor Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2					

Table A.15-1/X.862 – Commitment (sheet 50 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-REPORT (rollbackRI, completion-report) ind		^Dsup, Dcoor ^Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] 23.2					
	^Dsup, Dcoor Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2	^Dsup, Dcoor Du, ^Dcr Dcdfu [SETDIAGSB] [TRBi] [TREP] [CRBrs] [INITRB] [OWEDONE] [COUNTRB] [CPSAP] 23.2					
AF-ABORT-AND-REPORT (rollbackRI, heuristic-report) ind	^Dsup, Dcoor ^Dtb, Du, ^Dcr ^Dcdfu, ^Dhrsfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, Dcoor ^Dtb, ^Dcr ^Dcdfu, ^Dhrsfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2					

(Continued on sheet 51 of 87)

Table A.15-1/X.862 – Commitment (sheet 51 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT-AND-REPORT (rollbackRI, heuristic-report) ind *(Concluded 2 of 2)*	^Dsup, Dcoor Dtb, Du, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, Dcoor Dtb, ^Dcr ^Dhrsfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2					
AF-ABORT-AND-REPORT (rollbackRI, heuristic-report, completion-report) ind *(Continued on sheet 52 of 87)*	^Dsup, Dcoor ^Dtb, Du, ^Dcr ^Dhrsfu Dcdfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, Dcoor ^Dtb, ^Dcr ^Dhrsfu Dcdfu [TUABiR] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2					

Table A.15-1/X.862 – Commitment (sheet 52 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT-AND-REPORT (rollbackRI, heuristic-report, completion-report) ind *(Concluded 2 of 2)*	^Dsup, Dcoor Dtb, Du, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, Dcoor Dtb, ^Dcr ^Dhrsfu Dcdfu [SETDIAGSB] [TRBi] [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2					
AF-ABORT-AND-REPORT (rollbackRI, completion-report) ind *(Continued on sheet 53 of 87)*	^Dsup, Dcoor ^Dtb, Du, ^Dcr Dhrsfu, Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, Dcoor ^Dtb, ^Dcr Dhrsfu, Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2					

Table A.15-1/X.862 – Commitment (sheet 53 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-ABORT-AND-REPORT (rollbackRI, completion-report) ind *(Concluded 2 of 2)*	^Dsup, Dcoor Dtb, Du, ^Dcr Dhrsfu, Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2	^Dsup, Dcoor Dtb, ^Dcr Dhrsfu, Dcdfu [TUABiR] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [INITRB] [OWEDONE] [COUNTRB] 23.2					
CAF-RECOVER (ready) ind *(Continued on sheet 54 of 87)*	^Dsup, ^Dcoor, ^Danyb ^Ldretry [DIALOGUE] [VDchatT] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDETR] [VNfaT] 20.3			^Dcoor, D2pc ^Danyb ^Dch [RECVRCOMR] [DIALOGUE] [VDchatT] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [OWEDONE] [COUNTCR] 21.1		^Dcoor, D2pc [CRErsRTSB] [CAFDETrqF] 21.3 Dcoor, ^D2pc, Db [CRErsU] [CAFDETrqF] 21.3	

Table A.15-1/X.862 – Commitment (sheet 54 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
CAF-RECOVER (ready) ind *(Continued 2 of 7)*	Dsup, ^Dcoor, ^Danyb ^Ldretry [DIALOGUE] [VDchatT] [SETDIAGTP] [TPABi] [CRNALL] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDETR] [VNfaT] 20.3			^Dcoor, D2pc ^Danyb Dch [RECVRCOMR] [DIALOGUE] [VDchatT] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [RBNEXTSB] [OWEDONE] [COUNTCR] 21.1	^Dcoor, D2pc ^Danyb [RECVRCOMR] [DIALOGUE] [VDchatT] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [OWEDONE] [COUNTCR] 21.1		
	^Dsup, ^Dcoor, ^Danyb Ldretry [CRErsRT] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDETR] [VNfaT] 20.3			^Dcoor, D2pc Danyb ^Db [RECVRCOMR] [DIALOGUE] [VDchatT] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [COUNTCR] 21.1	^Dcoor, D2pc Danyb ^Db [RECVRCOMR] [DIALOGUE] [VDchatT] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [COUNTCR] 21.1		
(Continued on sheet 55 of 87)	Dsup, ^Dcoor, ^Danyb Ldretry [CRErsRT] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDETR] [VNfaT] [CRNALL] 20.3			^Dcoor, D2pc Dchat [CRErsRTSB] [CAFDETrqF] [OLDCHANNEL] 21.1			

Table A.15-1/X.862 – Commitment (sheet 55 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
CAF-RECOVER (ready) ind *(Continued 3 of 7)*	^Dcoor, Danyb ^Db ^Ldretry [DIALOGUE] [VDchatT] [SETDIAG] [AABrqPa] [ABDETR] 20.3						
	^Dcoor, Danyb ^Db Ldretry [CRErsRT] [CAFDETrqF] [DIALOGUE] [SETDIAG] [AABrqPa] [ABDETR] 20.3						
	^Dcoor, Danyb Db, ^Dchat ^Ldretry [VDchatT] 20.3						
	^Dcoor, Danyb Db, ^Dchat Ldretry [CRErsRT] [CAFDETrqF] [OLDCHANNEL] 20.3						
(Continued on sheet 56 of 87)	^Dcoor, Danyb Db, Dchat ^Ldretryo [CRErsRT] [CAFDETrqF] [OLDCHANNEL] 20.3						

Table A.15-1/X.862 – Commitment (sheet 56 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
CAF-RECOVER (ready) ind *(Continued 4 of 7)*	^Dcoor, Danyb Db, Dchat Ldretryo ^Ldretry [OLDCHANNEL] [CRErsRT] [CAFDETrqF] 20.3						
	^Dcoor, Danyb Db, Dchat Ldretry Ldretryo [CRErsRT] [CAFDETrqF] [OLDCHANNEL] [CRErsRT] [CAFDETrqF] [VDchatF] 20.3						
(Continued on sheet 57 of 87)	Dsup, Dcoor, ^Danyb [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [LOGDAMRB] [INITRB] [OWEDONE] [CRNALL] [SETDIAG] [AABrqPa] [ABDET] [VNfaT] 23.8	Dsup, Dcoor, ^Danyb [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [LOGDAMRB] [INITRB] [OWEDONE] [CRNALL] [SETDIAG] [AABrqPa] [ABDET] [VNfaT] 23.8	Dsup, Dcoor, ^Danyb [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [LOGDAMRB] [INITRB] [OWEDONE] [CRNALL] [SETDIAG] [AABrqPa] [ABDET] [VNfaT] 23.8				

Table A.15-1/X.862 – Commitment (sheet 57 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
CAF-RECOVER (ready) ind *(Continued 5 of 7)*	Dsup, Dcoor, Danyb, ^Db [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TRBi] [LOGDAMRB] [INITRB] [OWEDONE] [SETDIAG] [AABrqPa] [ABDET] [VNfaT] 23.8	Dsup, Dcoor, Danyb [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TRBi] [LOGDAMRB] [INITRB] [OWEDONE] [SETDIAG] [AABrqPa] [ABDET] [VNfaT] 23.8	Dsup, Dcoor, Danyb [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TRBi] [LOGDAMRB] [INITRB] [OWEDONE] [SETDIAG] [AABrqPa] [ABDET] [VNfaT] 23.8				
(Continued on sheet 58 of 87)	^Dsup, Dcoor, ^Danyb [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [LOGDAMRB] [INITRB] [OWEDONE] [COUNTRB] [SETDIAG] [AABrqPa] [ABDET] [VNfaT] 23.2	^Dsup, Dcoor, ^Danyb [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABiR] [LOGDAMRB] [INITRB] [OWEDONE] [COUNTRB] [SETDIAG] [AABrqPa] [ABDET] [VNfaT] 23.2					

Table A.15-1/X.862 – Commitment (sheet 58 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
CAF-RECOVER (ready) ind *(Continued 6 of 7)*	^Dsup, Dcoor, Danyb, ^Db [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TRBi] [LOGDAMRB] [INITRB] [OWEDONE] [COUNTRB] [SETDIAG] [AABrqPa] [ABDET] [VNfaT] 23.2	^Dsup, Dcoor, Danyb [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TRBi] [LOGDAMRB] [INITRB] [OWEDONE] [COUNTRB] [SETDIAG] [AABrqPa] [ABDET] [VNfaT] 23.2					
(Continued on sheet 59 of 87)	^Dsup, Dcoor, Dchat [CRErsU] [CAFDETrqF] [OLDCHANNEL] [CRErsU] [CAFDETrqF] [SETDIAGTP] [TRBi] [LOGDAMRB] [INITRB] [OWEDONE] [COUNTRB] [VNfaT] [VDchatF] 23.2						

Table A.15-1/X.862 – Commitment (sheet 59 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
CAF-RECOVER (ready) ind *(Concluded 7 of 7)*	Dsup, Dcoor, Dchat [CRErsU] [CAFDETrqF] [OLDCHANNEL] [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TRBi] [LOGDAMRB] [INITRB] [OWEDONE] [COUNTRB] [VNfaT] [VDchatF] 23.8						
CAF-RECOVER (commit) ind *(Continued on sheet 60 of 87)*	Dsup, Dcoor,^Danyb Ldretry [CRErsRTSP] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [RECCOM] [OWEDONECO] [VNfaT] 21.5					Dcoor, ^Danyb Dhrsfu, ^Nclw, Ldretry [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [OWEDONE] 21.3	

Table A.15-1/X.862 – Commitment (sheet 60 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
CAF-RECOVER (commit) ind *(Continued 2 of 7)*	Danyb, ^Db Dsup, Dcoor, Ldretry [CRErsRTSP] [CAFDETrqF] [DIALOGUE] [VDanuF] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [RECCOM] [OWEDONECO] [VNfaT] 21.5					Dcoor, ^Danyb Dhrsfu, Nclw [VDchatT] [EARLYC] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [OWEDONE] 21.3	
	Dsup, Dcoor, ^Danyb ^Ldretry [DIALOGUE] [VDchatT] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [NOTCHAIN] [RECCOM] [OWEDONECO] [VNfaT] 21.5					Dcoor Danyb, ^Db Dhrsfu, ^Nclw, Ldretry [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [SETDIAG] [AABrqPa] [ABDET] 21.3	
(Continued on sheet 61 of 87)						Dcoor Danyb, ^Db Dhrsfu, Nclw [VDchatT] [EARLYC] [DIALOGUE] [SETDIAG] [AABrqPa] [ABDET] 21.3	

Table A.15-1/X.862 – Commitment (sheet 61 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
CAF-RECOVER (commit) ind *(Continued 3 of 7)*	Danyb, ^Db Dsup, Dcoor, ^Ldretry [DIALOGUE] [VDchatT] [VDanuF] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [NOTCHAIN] [RECCOM] [OWEDONECO] 21.5					Dcoor Db, ^Dchat Dhrsfu, ^Nclw, Ldretry [CRErsRTSB] [CAFDETrqF] 21.3	
	Dsup, Dcoor, Dchat Ldretry [CRErsRTSP] [CAFDETrqF] [OLDCHANNEL] [CAFDETrqCU] [VDchatF] [RECCOM] [OWEDONECO] 21.5					Dcoor Db, ^Dchat Dhrsfu, Nclw [VDchatT] [EARLYC] 21.3	
	Dsup, Dcoor, Dchat ^Ldretry [OLDCHANNEL] [CAFDETrqCU] [NEWCHANNEL] [RECCOM] [OWEDONECO] 21.5					Dcoor, ^Danyb Dhrsfu, ^Nclw, ^Ldretry [DIALOGUE] [VDchatT] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [OWEDONE] 21.3	
(Continued on sheet 62 of 87)						Dcoor Danyb, ^Db Dhrsfu, ^Nclw, ^Ldretry [DIALOGUE] [VDchatT] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] 21.3	

Table A.15-1/X.862 – Commitment (sheet 62 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
CAF-RECOVER (commit) ind *(Continued 4 of 7)*	^Dsup, Dcoor, ^Danyb Dhrsfu, Ldretry [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [RECCOM] [OWEDONECO] [VNfaT] 21.3					Dcoor, Db, ^Dchat Dhrsfu, ^Nclw, ^Ldretry [VDchatT] 21.3	
						Dcoor, Dchat Dhrsfu, Ldretryo, Ldretry [CRErsRTSB] [CAFDETrqF] [OLDCHANNEL] [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [VDchatF] 21.3	
	^Dsup, Dcoor Danyb, ^Db Dhrsfu, Ldretry [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [SETDIAG] [AABrqPa] [ABDET] [RECCOM] [OWEDONECO] 21.3					Dcoor, Dchat Dhrsfu, Ldretryo, ^Ldretry [OLDCHANNEL] [CRErsRTSB] [CAFDETrqF] [NEWCHANNEL] 21.3	
(Continued on sheet 63 of 87)						Dcoor, Dchat Dhrsfu, ^Ldretryo [CRErsRTSB] [CAFDETrqF] [OLDCHANNEL] 21.3	

Table A.15-1/X.862 – Commitment (sheet 63 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
CAF-RECOVER (commit) ind *(Continued 5 of 7)*	^Dsup, Dcoor, ^Danyb Dhrsfu, ^Ldretry [DIALOGUE] [VDchatT] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [RECCOM] [OWEDONECO] [VNfaT] 21.3					Dcoor ^Danyb ^Dhrsfu [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [OWEDONE] 21.3	
	^Dsup, Dcoor Danyb, ^Db Dhrsfu, ^Ldretry [DIALOGUE] [VDchatT] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [RECCOM] [OWEDONECO] 21.3					Dcoor Danyb, ^Db ^Dhrsfu [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [VDanuF] [SETDIAG] [AABrqPa] [ABDET] 21.3	
(Continued on sheet 64 of 87)	^Dsup, Dcoor, Dchat Dhrsfu, Ldretry [CRErsRTSB] [CAFDETrqF] [OLDCHANNEL] [CAFDETrqCU] [DIALOGUE] [VDchatF] [RECCOM] [OWEDONECO] 21.3					Dcoor Db, ^Dchat ^Dhrsfu [CRErsRTSB] [CAFDETrqF] 21.3	

Table A.15-1/X.862 – Commitment (sheet 64 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
CAF-RECOVER (commit) ind *(Continued 6 of 7)*	^Dsup, Dcoor, Dchat Dhrsfu, ^Ldretry [OLDCHANNEL] [CAFDETrqCU] [NEWCHANNEL] [RECCOM] [OWEDONECO] 21.3					Dcoor, Dchat ^Dhrsfu [CRErsRTSB] [CAFDETrqF] [OLDCHANNEL] [CRErsRTSB] [CAFDETrqF] [VDchatF] 21.3	
	^Dsup, Dcoor ^Danyb ^Dhrsfu [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [RECCOM] [OWEDONECO] [VNfaT] 21.1			Dcoor, D2pc ^Danyb ^Dhrsfu [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [OWEDONE] [COUNTCR] 21.1			
(Continued on sheet 65 of 87)	^Dsup, Dcoor Danyb, ^Db ^Dhrsfu [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [VDanuF] [SETDIAG] [AABrqPa] [ABDET] [RECCOM] [OWEDONECO] 21.1			Dcoor, D2pc Danyb, ^Db ^Dhrsfu [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [VDanuF] [SETDIAG] [AABrqPa] [ABDET] [COUNTCR] 21.1			

Table A.15-1/X.862 – Commitment (sheet 65 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
CAF-RECOVER (commit) ind *(Concluded 7 of 7)*	^Dsup, Dcoor, Dchat ^Dhrsfu [CRErsRTSB] [CAFDETrqF] [OLDCHANNEL] [CAFDETrqCU] [VDchatF] [RECCOM] [OWEDONECO] 21.1			Dcoor Db, ^Dchat ^Dhrsfu [CRErsRTSB] [CAFDETrqF] 21.1			
CAF-RECOVER (commit, heuristic-report) ind *(Continued on sheet 66 of 87)*	^Dsup, Dcoor ^Danyb, ^Dhrsfu Ldretry [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [RECCOM] [TREP] [LOGDAM] [OWEDONECO] [VNfaT] [COUNTCOM] 21.3			Dcoor ^Danyb ^Dhrsfu ^Nclw, Ldretry [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [TREP] [LOGDAM] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		Dcoor ^Danyb ^Dhrsfu ^Nclw, Ldretry [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [OWEDONE] 21.3	

Table A.15-1/X.862 – Commitment (sheet 66 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
CAF-RECOVER (commit, heuristic-report) ind *(Continued 2 of 5)*	^Dsup, Dcoor Danyb, ^Db ^Dhrsfu Ldretry [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [SETDIAG] [AABrqPa] [ABDET] [RECCOM] [TREP] [LOGDAM] [OWEDONECO] [COUNTCOM] 21.3			Dcoor ^Danyb ^Dhrsfu Nclw [VDchatT] [EARLYC] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [TREP] [LOGDAM] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		Dcoor ^Danyb ^Dhrsfu Nclw [VDchatT] [EARLYC] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [OWEDONE] 21.3	
(Continued on sheet 67 of 87)	^Dsup, Dcoor ^Danyb ^Dhrsfu ^Ldretry [DIALOGUE] [VDchatT] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [RECCOM] [TREP] [LOGDAM] [OWEDONECO] [VNfaT] [COUNTCOM] 21.3			Dcoor Danyb, ^Db ^Dhrsfu ^Nclw, Ldretry [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [SETDIAG] [AABrqPa] [ABDET] [TREP] [LOGDAM] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		Dcoor Danyb, ^Db ^Dhrsfu ^Nclw, Ldretry [CRErsRTSB] [CAFDETrqF] [DIALOGUE] [SETDIAG] [AABrqPa] [ABDET] 21.3	

Table A.15-1/X.862 – Commitment (sheet 67 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
CAF-RECOVER (commit, heuristic-report) ind *(Continued 3 of 5)*	^Dsup, Dcoor Danyb, ^Db ^Dhrsfu ^Ldretry [DIALOGUE] [VDchatT] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [RECCOM] [TREP] [LOGDAM] [OWEDONECO] [COUNTCOM] 21.3			Dcoor Danyb, ^Db ^Dhrsfu, Nclw [VDchatT] [EARLYC] [DIALOGUE] [SETDIAG] [AABrqPa] [ABDET] [TREP] [LOGDAM] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		Dcoor Danyb, ^Db ^Dhrsfu, Nclw [VDchatT] [EARLYC] [DIALOGUE] [SETDIAG] [AABrqPa] [ABDET] 21.3	
	^Dsup, Dcoor, Dchat ^Dhrsfu Ldretry [CRErsRTSB] [CAFDETrqF] [OLDCHANNEL] [CAFDETrqCU] [VDchatF] [RECCOM] [TREP] [LOGDAM] [OWEDONECO] [COUNTCOM] 21.3			Dcoor, Db, ^Dchat ^Dhrsfu ^Nclw, Ldretry [CRErsRTSB] [CAFDETrqF] [TREP] [LOGDAM] [COUNTREP] [COUNTCOM] 21.3		Dcoor, Db, ^Dchat ^Dhrsfu ^Nclw, Ldretry [CRErsRTSB] [CAFDETrqF] 21.3	
(Continued on sheet 68 of 87)				Dcoor, Db, ^Dchat ^Dhrsfu Nclw [VDchatT] [EARLYC] [TREP] [LOGDAM] [COUNTREP] [COUNTCOM] 21.3		Dcoor, Db, ^Dchat ^Dhrsfu Nclw [VDchatT] [EARLYC] 21.3	

Table A.15-1/X.862 – Commitment (sheet 68 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
CAF-RECOVER (commit, heuristic-report) ind *(Continued 4 of 5)*	^Dsup, Dcoor, Dchat ^Dhrsfu ^Ldretry [OLDCHANNEL] [CAFDETrqCU] [NEWCHANNEL] [RECCOM] [TREP] [LOGDAM] [OWEDONECO] [COUNTCOM] 21.3			Dcoor ^Danyb ^Dhrsfu ^Nclw, ^Ldretry [DIALOGUE] [VDchatT] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [TREP] [LOGDAM] [OWEDONE] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		Dcoor ^Danyb ^Dhrsfu ^Nclw, ^Ldretry [DIALOGUE] [VDchatT] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [OWEDONE] 21.3	
(Continued on sheet 69 of 87)				Dcoor Danyb, ^Db ^Dhrsfu ^Nclw,^Ldretry [DIALOGUE] [VDchatT] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [TREP] [LOGDAM] [COUNTCR] [COUNTREP] [COUNTCOM] 21.3		Dcoor Danyb, ^Db ^Dhrsfu ^Nclw,^Ldretry [DIALOGUE] [VDchatT] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] 21.3	

Table A.15-1/X.862 – Commitment (sheet 69 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
CAF-RECOVER (commit, heuristic-report) ind *(Concluded 5 of 5)*				Dcoor, Db, ^Dchat ^Dhrsfu ^Nclw,^Ldretry [VDchatT] [TREP] [LOGDAM] [COUNTREP] [COUNTCOM] 21.3		Dcoor, Db, ^Dchat ^Dhrsfu ^Nclw,^Ldretry [VDchatT] 21.3	
						Dcoor, Dchat Dhrsfu, Ldretryo, Ldretry [CRErsRTSB] [CAFDETrqF] [OLDCHANNEL] [CRErsRTSB] [CAFDETrqF] [VDchatF] 21.3	
						Dcoor, Dchat ^Dhrsfu Ldretryo, ^Ldretry [OLDCHANNEL] [CRErsRTSB] [CAFDETrqF] [NEWCHANNEL] 21.3	
						Dcoor, Dchat ^Dhrsfu, ^Ldretryo [CRErsRTSB] [CAFDETrqF] [OLDCHANNEL] 21.3	
C-RECOVER (commit) ind *(Continued on sheet 70 of 87)*	Dsup, Dcoor, Dchat Ldretry [CRErsRTSP] [CAFDETrqF] [VDchatF] [RECCOM] [OWEDONECO] 21.5						

Table A.15-1/X.862 – Commitment (sheet 70 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
C-RECOVER (commit) ind *(Concluded 2 of 2)*	Dsup, Dcoor, Dchat ^Ldretry [RECCOM] [OWEDONECO] 21.5						
	^Dsup, Dcoor, Dchat Dhrsfu, Ldretry [CRErsRTSB] [CAFDETrqF] [VDchatF] [RECCOM] [OWEDONECO] 21.3						
	^Dsup, Dcoor, Dchat Dhrsfu, ^Ldretry [RECCOM] [OWEDONECO] 21.3						
	^Dsup, Dcoor, Dchat ^Dhrsfu [CRErsRTSB] [CAFDETrqF] [VDchatF] [RECCOM] [OWEDONECO] 21.1						
AF-REPORT(commitRI, heuristic-report) ind *(Continued on sheet 71 of 87)*	^Dsup, Dcoor Dchat ^Ldretry ^Dhrsfu [RECCOM] [TREP] [LOGDAM] [OWEDONECO] [COUNTREP] [COUNTCOM] 21.3			Dcoor, D2pc Dchat ^Dhrsfu ^Nclw, ^Ldretry [TREP] [LOGDAM] [COUNTREP] [COUNTCOM] 21.3			

Table A.15-1/X.862 – Commitment (sheet 71 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-REPORT(commitRI, heuristic-report) ind *(Concluded 2 of 2)*	^Dsup, Dcoor, Dchat Ldretry ^Dhrsfu [CRErsRTSB] [CAFDETrqF] [VDchatF] [RECCOM] [TREP] [LOGDAM] [OWEDONECO] [COUNTREP] [COUNTCOM] 21.3			Dcoor, D2pc Dchat ^Dhrsfu ^Nclw, Ldretry [CRErsRTSB] [CAFDETrqF] [VDchatF] [TREP] [LOGDAM] [COUNTREP] [COUNTCOM] 21.3			
				Dcoor, D2pc Dchat ^Dhrsfu Nclw [TREP] [LOGDAM] [COUNTREP] [COUNTCOM] [EARLYC] 21.3			
C-RECOVER (done) cnf				^Dcoor, ^Atwr [CAFDETrqF] [VDchatF] [COUNTREP] [COUNTCOM] 21.3			
				^Dcoor, Atwr, ^Atokx [CAFDETrqF] [VDchatF] [COUNTREP] [COUNTCOM] 21.3			

Table A.15-1/X.862 – Commitment (sheet 72 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
AF-REPORT (recoverDoneRC) ind				^Dcoor, ^Dhrsfu, ^Atwr [CAFDETrqF] [TREP] [LOGDAM] [VDchatF] [COUNTREP] [COUNTCOM] 21.3			
				^Dcoor, ^Dhrsfu, Atwr, ^Atokx [CAFDETrqF] [TREP] [LOGDAM] [VDchatF] [COUNTREP] [COUNTCOM] 21.3			
C-RECOVER (unknown) cnf	Dsup [CAFDETrqF] [SETDIAGSP] [TRBi] [LOGDAMRB] [VDchatF] [INITRB] [OWEDONE] 23.8						
	^Dsup [CAFDETrqF] [SETDIAGSB] [TRBi] [LOGDAMRB] [VDchatF] [INITRB] [OWEDONE] [COUNTRB] 23.2						

Table A.15-1/X.862 – Commitment (sheet 73 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
C-RECOVER (retry-later) cnf	^Atwr [CAFDETrqF] [VDchatF] [VDrvypT] 99			^Dcoor, ^Atwr [CAFDETrqF] [VDchatF] [VDrvypT] 99			
	Atwr, ^Atokx [CAFDETrqF] [VDchatF] [VDrvypT] 99			^Dcoor, Atwr, ^Atokx [CAFDETrqF] [VDchatF] [VDrvypT] 99			
AF-TOKEN-GIVE (two-way-recovery) ind	Atwr, Atokx ^Dsup, Dchat [VAtokxF] 20.3			Atwr, Atokx Dchat [VAtokxF] 21.1			
AF-TOKEN-PLEASE ind	Dchat 20.3			Dchat 21.1			
CAF-GIVE ind				[CAFDETrqNU] 21.1		[CAFDETrqNU] 21.3	
CAF-FAIL ind				21.1		21.3	
Heuristic-decision	Dsup [LOGHD] 20.3						
	^Dsup, Ni 20.3						
Heuristic-damage-comp	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 20.3			memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 21.1	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 21.2	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 21.3	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 21.4
Delay-recovery	^Dcoor, Dchat [CRErsRT] [CAFDETrqF] [VDchatF] 20.3					Dcoor, Dchat [CRErsRTSB] [CAFDETrqF] [VDchatF] 21.3	

Table A.15-1/X.862 – Commitment (sheet 74 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
Restart-TPPM	Dsup, Dcoor [CAFPLrqSP] 99			[CAFPLrqSB] 99			
	^Dsup, Dcoor [CAFPLrqSB] 99						
	^Dcoor 20.3						
Rollback-by-TPPM	Dsup ^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [CANCEL] 23.3						
	Dsup Nfrb [CANCEL] 23.3						
	^Dsup ^Nfrb [SETDIAGLO] [TRBi] [RBREQ] [INITRB] [OWEDONE] 23.1						
	^Dsup Nfrb [RBREQ] 23.1						
Rollback-all	^Dcoor, Dsup, ^Db [CANCEL] 23.3	^Dcoor,Dsup, ^Db [CANCEL] 23.3					
	^Dcoor, Dsup, Db ^Dchat 23.8	^Dcoor,Dsup, Db ^Dchat 23.8					

(Continued on sheet 75 of 87)

Table A.15-1/X.862 – Commitment (sheet 75 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
Rollback-all *(Concluded 2 of 2)*	^Dcoor, Dsup, Dchat [CRErsU] [CAFDETrqF] [VDchatF] 23.8						
	^Dcoor, ^Dsup, ^Deei, ^Db [RBREQ] 23.1	^Dcoor,^Dsup, ^Deei, ^Db [RBREQ] 23.1	^Dcoor,^Dsup, ^Deei, ^Db [RBREQ] 23.1				
	^Dcoor, ^Dsup, ^Deei, Db ^Dchat [LOGDAMH] [TREP] [COUNTRB] 23.2	^Dcoor,^Dsup, ^Deei, Db ^Dchat [COUNTRB] 23.2	^Dcoor,^Dsup, ^Deei, Db ^Dchat [COUNTRB] 23.2				
	^Dcoor,^Dsup, Dchat [LOGDAMH] [TREP] [CRErsU] [CAFDETrqF] [VDchatF] [COUNTRB] 23.2						
	^Dcoor,^Dsup, Deei [COUNTRB] 23.2	^Dcoor,^Dsup, Deei [COUNTRB] 23.2	^Dcoor,^Dsup, Deei [COUNTRB] 23.2				
Set-done-true	[VDdT] 20.3	[VDdT] 20.3.2	[VDdT] 20.3.3	[VDdT] 21.1	[VDdT] 21.2	[VDdT] 21.3	[VDdT] 21.4
cr-allowed	[VDcrpaT] 20.3	[VDcrpaT] 20.3.2	[VDcrpaT] 20.3.3	[VDcrpaT] 21.1	[VDcrpaT] 21.2	[VDcrpaT] 21.3	[VDcrpaT] 21.4

Table A.15-1/X.862 – Commitment (sheet 76 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
report-status				^Dhrsfu, ^Dcdfu Dnchra [VNhrstF] [VNcnthrINC] 21.1 ^Db, ^Dhrsfu, Dcdfu Dnchra [VNhrstF] [VNcnthrINC] [VNcrstF] [VNcntcrINC] 21.1		21.3	
				Db, ^Dhrsfu, Dcdfu Dnchra [VNhrstF] [VNcnthrINC] 21.1			
				^Db, Dhrsfu, Dcdfu [VNcrstF] [VNcntcrINC] 21.1			
				Db, Dhrsfu, Dcdfu 21.1 Dhrsfu, ^Dcdfu 21.1			
				^Db, ^Dhrsfu, ^Dnchra Dcdfu [VNcrstF] [VNcntcrINC] 21.1			
				Db, ^Dhrsfu, ^Dnchra Dcdfu 21.1			
				^Dhrsfu, ^Dnchra ^Dcdfu 21.1			

Table A.15-1/X.862 – Commitment (sheet 77 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
Continue-commit	^Dsup, Drdyi, ^Db ^De Ptok [COMREQ] 21.1			Dcoor 21.1		Dcoor 21.3	
	^Dsup, Drdyi, ^Db De Ptok [NOTCHAIN] [COMREQ] 21.1						
	^Dsup, Dchat [VD2pcT] [RECVRCOMR] [VNcntINC] 21.1						
	^Dsup, Drdyi, ^Dchat Db [VD2pcT] [VNcntINC] [CAFPLrqSB] 99						
	^Dsup, Dopi, Du ^Db, ^De, ^Dtb [COMREQ] [CPSAP] 21.3	^Dsup, Dopi, Du ^Db, ^De, ^Dtb [COMREQ] [CPSAP] 21.3					
(Continued on sheet 78 of 87)	^Dsup, Dopi, Du ^Db, ^De, Dtb [COMREQ] [SDETrqF] [ABDET] 21.3	^Dsup, Dopi, Du ^Db, ^De, Dtb [COMREQ] [SDETrqF] [ABDET] 21.3					

Table A.15-1/X.862 – Commitment (sheet 78 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
Continue-commit *(Continued 2 of 7)*	^Dsup, Dopi, Du ^Db, De [COMREQ] [SDETrqF] [ABDET] 21.3	^Dsup, Dopi, Du ^Db, De [COMREQ] [SDETrqF] [ABDET] 21.3					
	^Dsup, Dopi, Du Db 21.3	^Dsup, Dopi, Du Db 21.3					
	^Dsup, Droi, ^Db ^Dch, ^De, ^Dtb [COMREQ] [CPSAP] 21.3	^Dsup, Droi, ^Db ^Dch, ^De, ^Dtb [COMREQ] [CPSAP] 21.3	^Dsup, Droi, ^Db ^Dch, ^De, ^Dtb [COMREQ] [CPSAP] 21.3				
	^Dsup, Droi, ^Db ^De, Dtb [NOTCHAIN] [COMREQ] [SDETrqF] [ABDET] 21.3	^Dsup, Droi, ^Db ^De, Dtb [NOTCHAIN] [COMREQ] [SDETrqF] [ABDET] 21.3	^Dsup, Droi, ^Db ^De, Dtb [NOTCHAIN] [COMREQ] [SDETrqF] [ABDET] 21.3				
	^Dsup, Droi, ^Db De [NOTCHAIN] [COMREQ] [SDETrqF] [ABDET] 21.3	^Dsup, Droi, ^Db De [NOTCHAIN] [COMREQ] [SDETrqF] [ABDET] 21.3	^Dsup, Droi, ^Db De [NOTCHAIN] [COMREQ] [SDETrqF] [ABDET] 21.3				
	^Dsup, Droi, ^Db Dch, ^De, ^Dtb 21.3	^Dsup, Droi, ^Db Dch, ^De, ^Dtb 21.3	^Dsup, Droi, ^Db Dch, ^De, ^Dtb 21.3				
	^Dsup, Droi, Db 21.3	^Dsup, Droi, Db 21.3	^Dsup, Droi, Db 21.3				
(Continued on sheet 79 of 87)	^Dsup, Deei 21.3	^Dsup, Deei 21.3	^Dsup, Deei 21.3				

Table A.15-1/X.862 – Commitment (sheet 79 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
Continue-commit *(Continued 3 of 7)*	Dsup, Drdyi ^Db, ^De Ptok Dhrsfu, ^Dcdfu [COMREQ] 21.5.2						
	Dsup, Drdyi ^Db, ^De Ptok Dhrsfu, Dcdfu [COMREQ] [INITREPSP] 21.5.1						
	Dsup, Drdyi ^Db, ^De Ptok ^Dhrsfu, ^Dcdfu [COMREQ] [INITREPSP] 21.5.1						
	Dsup, Drdyi ^Db, ^De Ptok ^Dhrsfu, Dcdfu [COMREQ] [INITREPSP] 21.5.1						
(Continued on sheet 80 of 87)	Dsup, Drdyi ^Db, De Ptok Dhrsfu, ^Dcdfu [NOTCHAIN] [COMREQ] 21.5.2						

Table A.15-1/X.862 – Commitment (sheet 80 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
Continue-commit *(Continued 4 of 7)*	Dsup, Drdyi ^Db, De Ptok Dhrsfu, Dcdfu [NOTCHAIN] [COMREQ] [INITREPSP] 21.5.1						
	Dsup, Drdyi ^Db, De Ptok ^Dhrsfu, ^Dcdfu [NOTCHAIN] [COMREQ] [INITREPSP] 21.5.1						
	Dsup, Drdyi ^Db, De Ptok ^Dhrsfu, Dcdfu [NOTCHAIN] [COMREQ] [INITREPSP] 21.5.1						
	Dsup, Dchat Dhrsfu [VD2pcT] [RECVRCOMR] [VNcntINC] 21.5.2						
(Continued on sheet 81 of 87)	Dsup, Dchat ^Dhrsfu [RECVRCOMR] [VNcntINC] [VD2pcT] [INITREPSP] 21.5.1						

Table A.15-1/X.862 – Commitment (sheet 81 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
Continue-commit *(Continued 5 of 7)*	Dsup, Drdyi, ^Dchat Db, Dhrsfu [VD2pcT] [VNcntINC] [CAFPLrqSB] 99						
	Dsup, Drdyi, ^Dchat Db, ^Dhrsfu [VD2pcT] [VNcntINC] [CAFPLrqSB] [INITREPSP] 99						
	Dsup, Dopi, ^Db ^Dch, ^De, ^Dtb Dhrsfu, ^Dcdfu [COMREQ] [CPSAP] 21.5.4	Dsup, Dopi, ^Db ^Dch, ^De, ^Dtb Dhrsfu, ^Dcdfu [COMREQ] [CPSAP] 21.5.4					
	Dsup, Dopi, ^Db ^Dch, ^De, ^Dtb Dhrsfu, Dcdfu [COMREQ] [INITREPSP] 21.5.1	Dsup, Dopi, ^Db ^Dch, ^De, ^Dtb Dhrsfu, Dcdfu [COMREQ] [INITREPSP] 21.5.1					
	Dsup, Dopi, ^Db ^Dch, ^De, ^Dtb ^Dhrsfu [COMREQ] [INITREPSP] 21.5.1	Dsup, Dopi, ^Db ^Dch, ^De, ^Dtb ^Dhrsfu [COMREQ] [INITREPSP] 21.5.1					
(Continued on sheet 82 of 87)	Dsup, Dopi, ^Db ^De, Dtb Dhrsfu, ^Dcdfu [COMREQ] [SDETrqF] [ABDET] 21.5.4	Dsup, Dopi, ^Db ^De, Dtb Dhrsfu, ^Dcdfu [COMREQ] [SDETrqF] [ABDET] 21.5.4					

Table A.15-1/X.862 – Commitment (sheet 82 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
Continue-commit *(Continued 6 of 7)*	Dsup, Dopi, ^Db ^De, Dtb Dhrsfu, Dcdfu [COMREQ] [INITREPSP] 21.5.1	Dsup, Dopi, ^Db ^De, Dtb Dhrsfu, Dcdfu [COMREQ] [INITREPSP] 21.5.1					
	Dsup, Dopi, ^Db ^De, Dtb ^Dhrsfu [COMREQ] [INITREPSP] 21.5.1	Dsup, Dopi, ^Db ^De, Dtb ^Dhrsfu [COMREQ] [INITREPSP] 21.5.1					
	Dsup, Dopi, ^Db De Dhrsfu, ^Dcdfu [NOTCHAIN] [COMREQ] [SDETrqF] [ABDET] 21.5.4	Dsup, Dopi, ^Db De Dhrsfu, ^Dcdfu [NOTCHAIN] [COMREQ] [SDETrqF] [ABDET] 21.5.4					
	Dsup, Dopi, ^Db De Dhrsfu [NOTCHAIN] [COMREQ] [INITREPSP] 21.5.1	Dsup, Dopi, ^Db De Dhrsfu [NOTCHAIN] [COMREQ] [INITREPSP] 21.5.1					
(Continued on sheet 83 of 87)	Dsup, Dopi, ^Db De ^Dhrsfu, ^Dcdfu [NOTCHAIN] [COMREQ] [INITREPSP] 21.5.1	Dsup, Dopi, ^Db De ^Dhrsfu, ^Dcdfu [NOTCHAIN] [COMREQ] [INITREPSP] 21.5.1					

Table A.15-1/X.862 – Commitment (sheet 83 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
Continue-commit *(Concluded 7 of 7)*	Dsup, Dopi, ^Db Dch, ^De, ^Dtb Dhrsfu, ^Dcdfu [COMREQ] 21.5.3	Dsup, Dopi, ^Db Dch, ^De, ^Dtb Dhrsfu, ^Dcdfu [COMREQ] 21.5.3					
	Dsup, Dopi, ^Db Dch, ^De, ^Dtb Dhrsfu, Dcdfu [COMREQ] [INITREPSP] 21.5.1	Dsup, Dopi, ^Db Dch, ^De, ^Dtb Dhrsfu, Dcdfu [COMREQ] [INITREPSP] 21.5.1					
	Dsup, Dopi, ^Db Dch, ^De, ^Dtb ^Dhrsfu [COMREQ] [INITREPSP] 21.5.1	Dsup, Dopi, ^Db Dch, ^De, ^Dtb ^Dhrsfu [COMREQ] [INITREPSP] 21.5.1					
	Dsup, Dopi, Db 21.5.4	Dsup, Dopi, Db 21.5.4					
continue-unknown		^Dsup, Dopi, Du ^Db, ^De, ^Dtb [COMREQ] [CPSAP] 21.3				Dcoor 21.3	
		^Dsup, Dopi, Du ^Db, ^De, Dtb [COMREQ] [SDETrqF] [ABDET] 21.3					
		^Dsup, Dopi, Du ^Db, De [COMREQ] [SDETrqF] [ABDET] 21.3					
(Continued on sheet 84 of 87)		^Dsup, Dopi, Du Db 21.3					

Table A.15-1/X.862 – Commitment (sheet 84 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
continue-unknown *(Continued 2 of 3)*		^Dsup, Droi, ^Db ^Dch, ^De, ^Dtb [COMREQ] [CPSAP] 21.3	^Dsup, Droi, ^Db ^Dch, ^De, ^Dtb [COMREQ] [CPSAP] 21.3				
		^Dsup, Droi, ^Db ^De, Dtb [NOTCHAIN] [COMREQ] [SDETrqF] [ABDET] 21.3	^Dsup, Droi, ^Db ^De, Dtb [NOTCHAIN] [COMREQ] [SDETrqF] [ABDET] 21.3				
		^Dsup, Droi, ^Db De [NOTCHAIN] [COMREQ] [SDETrqF] [ABDET] 21.3	^Dsup, Droi, ^Db De [NOTCHAIN] [COMREQ] [SDETrqF] [ABDET] 21.3				
		^Dsup, Droi, ^Db Dch, ^De, ^Dtb 21.3	^Dsup, Droi, ^Db Dch, ^De, ^Dtb 21.3				
		^Dsup, Droi, Db 21.3	^Dsup, Droi, Db 21.3				
		^Dsup, Deei 21.3	^Dsup, Deei 21.3				
		Dsup, Dopi, ^Db ^Dch, ^De, ^Dtb [COMREQ] [CPSAP] 21.5.4					
(Continued on sheet 85 of 87)		Dsup, Dopi, ^Db ^De, Dtb [COMREQ] [SDETrqF] [ABDET] 21.5.4					

Table A.15-1/X.862 – Commitment (sheet 85 of 87)

State	20.3	20.3.2	20.3.3	21.1	21.2	21.3	21.4
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates Event	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
continue-unknown *(Concluded 3 of 3)*		Dsup, Dopi, ^Db De [NOTCHAIN] [COMREQ] [SDETrqF] [ABDET] 21.5.4					
		Dsup, Dopi, ^Db Dch, ^De, ^Dtb [COMREQ] 21.5.3					
		Dsup, Dopi, Db 21.5.4					
rewrite-log				Dcoor, D2pc [REWRLOG] 21.1		Dcoor, D2pc [REWRLOG] 21.3	
log-rewritten				Dcoor, D2pc 21.1		Dcoor, D2pc [EARLYC] 21.3	
				^Dcoor 21.1	21.2	^Dcoor 21.3	21.4
send-report				21.1	21.2	21.3	21.4
Rollback-next-trans				^Db, Dch, Dcancr 21.2		^Danyb, D2pc, Dch [CRBrq] 21.4	
				^Db, Dch, ^Dcancr [CANCEL] 21.2		^Danyb, ^D2pc, Dch, Dcancr [CANCEL] 21.4	
				Db, Dch 21.1		Db, Dch 21.3	
				^Dch 21.1		^Dch 21.3	

Table A.15-1/X.862 – Commitment (sheet 86 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
Complete-commit						Danyb, ^Danu ^Nrpend [CMPCOMSB] [DELBR] [CMPCOM] 1	D2pc [CMPCOMSB] [RESETD] [CMPCOM] 23.1
						Danyb, ^Danu Nrpend [CMPCOMSB] [COUNTRB] [CMPCOM] 23.2	^D2pc, ^Dcoor [RESETD] [COUNTRB] [CMPCOM] 23.2
						Danyb, Danu ^Nrpend [VDanuF] [CMPCOMSB] [DELNBRANCH] [CMPCOM] 25	Nr, ^D2pc, Dcoor [CMPCOMSB] [RBREQ] [RESETD] [CMPCOM] 23.1
						Danyb, Danu Nrpend [VDanuF] [CMPCOMSB] [COUNTRB] [CMPCOM] 23.2	
						^Danyb, De [CMPCOMSB] [DELBR] [CMPCOM] 1	
(Continued on sheet 87 of 87)						^Danyb, ^De Dch, Dsh [CMPCOMSB] [RESETD] [CMPCOM] 2	

Table A.15-1/X.862 – Commitment (sheet 87 of 87)

State	**20.3**	**20.3.2**	**20.3.3**	**21.1**	**21.2**	**21.3**	**21.4**
Node state	READY	ONE-PHASE	READONLY	DECIDED (commit) or DECIDED (unknown)			
	Decision awaited	Decision awaited	or EARLY-EXIT decision awaited	commit confirm (^Dcoor, D2pc) or report on data (Dcoor) awaited	commit confirm or report on data (static one-phase) awaited need to rollback	commit confirm received or not awaited, psap closed or Db if not Dcoor	commit confirm received or not awaited rollback initiated
Predicates **Event**	Dl	Dl	Dl	Dl ^Dsup	Dl, ^Dsup, Dch	Dl,^Dsup	Dl, ^Dsup, Dch
Complete-commit *(Concluded 2 of 2)*						^Danyb, ^De Dch, ^Dsh, Dg [CMPCOMSB] [RESETD] [CMPCOM] 3	
						^Danyb, ^De Dch, ^Dsh, ^Dg [CMPCOMSB] [RESETD] [CMPCOM] 2	
						^Danyb, ^De ^Dch, Dsh [CMPCOMSB] [DELBR] [RESETD] [CMPCOM] 2	
						^Danyb, ^De ^Dch, ^Dsh, Dg [CMPCOMSB] [DELBR] [RESETD] [CMPCOM] 3	
						^Danyb, ^De ^Dch, ^Dsh, ^Dg [CMPCOMSB] [DELBR] [RESETD] [CMPCOM] 2	

End of Table A.15-1/X.862

Table A.15-2/X.862 – Commitment (sheet 1 of 21)

State	**21.5**	**21.5.1**	**21.5.2**	**21.5.3**	**21.5.4**	**21.6**	**21.6.1**	**21.6.3**	**21.6.4**	**99**
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates **Event**	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
0TP-U-ABORT req	^Danyb Nfa, Nch [ABTPSUI] [RBNEXTSB] 21.5	^Danyb Nfa, Nch [ABTPSUI] [RBNEXTSB] 21.5.1	^Danyb Nfa [ABTPSUI] 21.5.2	^Danyb Nfa, Nch [ABTPSUI] [AABrqUd] [SDETrqBF] [ABDET] [RBNEXTSB] 21.5.3	^Danyb Nfa, Nch [ABTPSUI] [AABrqUr] [SDETrqRBC] [ABDET] [RBNEXTSB] 21.5.4	^Danyb Nfa [ABTPSUI] 21.6	^Danyb, Nfa [ABTPSUI] 21.6.1	^Danyb, Nfa [ABTPSUI] 21.6.3	^Danyb Nfa [ABTPSUI] 21.6.4	
	^Ncc, ^Danyb Nfa, ^Nch [ABTPSUI] 21.5	^Danyb Nfa, ^Nch [ABTPSUI] 21.5.1			^Danyb Nfa, ^Nch [ABTPSUI] [OPSAP] [AABrqUd] [SDETrqBF] [ABDET] 21.5.4					
	Ncc, ^Danyb Nfa, ^Nch [ABTPSUI] [AABrqUd] [SDETrqBF] [ABDET] 21.5									
AF-ABORT (user, dataRI) ind				^Danyb [TUABi] [ABDET] [SDETrqF] [COUNTCOM] [RBNEXTSB] 21.5.4						
				Danyb [ABDET] [SDETrqF] [COUNTCOM] [RBNEXTSB] 21.5.4						

Table A.15-2/X.862 – Commitment (sheet 2 of 21)

State	**21.5**	**21.5.1**	**21.5.2**	**21.5.3**	**21.5.4**	**21.6**	**21.6.1**	**21.6.3**	**21.6.4**	**99**
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates **Event**	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
AF-ABORT (provider, abortRI) ind or A-P-ABORT ind or A-ABORT ind or A-ABORT req or A-RELEASE (result = affirmative) rsp or A-RELEASE (result = affirmative) cnf	^Danyb, ^Nch [SETDIAGTP] [TPABi] [ABDET] [CRNALL] [OWEDONE] 21.5	^Danyb, D2pc [SETDIAGTP] [TPABi] [ABDET] [VNrpdcrF] [CRNALL] [OWEDONE] [CAFPLrqSP] 99	^Danyb, D2pc [SETDIAGTP] [TPABi] [ABDET] [CRNALL] [OWEDONE] [CAFPLrqSP] 99			^Danyb [SETDIAGTP] [TPABi] [ABDET] [CRNALL] [OWEDONE] 21.5				
		^Danyb, ^D2pc ^Nch [SETDIAGTP] [TPABi] [ABDET] [VNrpdcrF] [VNrpdhrF] [CRNALL] [OWEDONE] 21.5.4								
	Danyb ^Db [ABDET] 21.5	Danyb, D2pc ^Db [ABDET] [VNrpdcrF] [CAFPLrqSP] 99	Danyb, D2pc ^Db [ABDET] [VNrpdcrF] [CAFPLrqSP] 99			Danyb [ABDET] 21.5				
		Danu, ^D2pc [ABDET] [VNrpdcrF] [VNrpdhrF] 21.5.4					Danyb [ABDET] [VNrpdcrF] [VNrpdhrF] [COUNTCOM] 21.5.4	Danyb [ABDET] [VNrpdcrF] [VNrpdhrF] [COUNTCOM] 21.5.4		
(Continued on sheet 3 of 21)		Dtb, ^Danu, ^D2pc [ABDET] [VNrpdcrF] [VNrpdhrF] 21.5.4								

Table A.15-2/X.862 – Commitment (sheet 3 of 21)

State	**21.5**	**21.5.1**	**21.5.2**	**21.5.3**	**21.5.4**	**21.6**	**21.6.1**	**21.6.3**	**21.6.4**	**99**
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates **Event**	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
AF-ABORT (provider, abortRI) ind or A-P-ABORT ind or A-ABORT ind or A-ABORT req or A-RELEASE (result = affirmative) rsp or A-RELEASE (result = affirmative) cnf *(Concluded 2 of 2)*	Dchat [VDchatF] 21.5		Dchat D2pc [VDchatF] [CAFPLrqSP] 99							
	^Danyb, Nch [SETDIAGTP] [TPABi] [ABDET] [CRNALL] [OWEDONE] [RBNEXTSB] 21.5	^Danyb, Nch ^D2pc [SETDIAGTP] [TPABi] [ABDET] [VNrpdcrF] [VNrpdhrF] [CRNALL] [OWEDONE] [RBNEXTSB] [COUNTCOM] 21.5.4		[SETDIAGTP] [TPABi] [ABDET] [CRNALL] [OWEDONE] [RBNEXTSB] [COUNTCOM] 21.5.4			^Danyb [SETDIAGTP] [TPABi] [ABDET] [VNrpdcrF] [VNrpdhrF] [CRNALL] [OWEDONE] [COUNTCOM] 21.5.4	^Danyb [SETDIAGTP] [TPABi] [ABDET] [CRNALL] [OWEDONE] [COUNTCOM] 21.5.4		
Protocol error or Internal error	^Danyb, ^Nch [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [CRNALL] [OWEDONE] 21.5	^Danyb, D2pc [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [VNrpdcrF] [CRNALL] [OWEDONE] [CAFPLrqSP] 99	^Danyb, D2pc [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [CRNALL] [OWEDONE] [CAFPLrqSP] 99		^Danyb, ^Nch [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [OWEDONE] 21.5.4	^Danyb [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [CRNALL] [OWEDONE] 21.5				
(Continued on sheet 4 of 21)		^Danyb, ^D2pc ^Nch [SETDIAGTP] [TPABi] [ABDET] [VNrpdcrF] [VNrpdhrF] [CRNALL] [OWEDONE] 21.5.4								

Table A.15-2/X.862 – Commitment (sheet 4 of 21)

State	**21.5**	**21.5.1**	**21.5.2**	**21.5.3**	**21.5.4**	**21.6**	**21.6.1**	**21.6.3**	**21.6.4**	**99**
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates **Event**	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
Protocol error or Internal error *(Continued 2 of 3)*	Danyb ^Db [ABDET] [SETDIAG] [AABrqPa] 21.5	Danyb, D2pc ^Db [ABDET] [SETDIAG] [AABrqPa] [VNrpdcrF] [CAFPLrqSP] 99	Danyb, D2pc ^Db [ABDET] [SETDIAG] [AABrqPa] [VNrpdcrF] [CAFPLrqSP] 99		Danyb ^Db [ABDET] [SETDIAG] [AABrqPa] 21.5.4	Danyb [ABDET] [SETDIAG] [AABrqPa] 21.5				
		Danu, ^D2pc [ABDET] [SETDIAG] [AABrqPa] [VNrpdcrF] [VNrpdhrF] 21.5.4					Danyb [ABDET] [SETDIAG] [AABrqPa] [VNrpdcrF] [VNrpdhrF] [COUNTCOM] 21.5.4	Danyb [ABDET] [SETDIAG] [AABrqPa] [VNrpdcrF] [VNrpdhrF] [COUNTCOM] 21.5.4		
		Dtb, ^Danu, ^D2pc [ABDET] [SETDIAG] [AABrqPa] [VNrpdcrF] [VNrpdhrF] 21.5.4								
	Dchat [SETDIAG] [AABrqPa] [VDchatF] 21.5		Dchat D2pc [SETDIAG] [AABrqPa] [VDchatF] [CAFPLrqSP] 99							

Table A.15-2/X.862 – Commitment (sheet 5 of 21)

State	**21.5**	**21.5.1**	**21.5.2**	**21.5.3**	**21.5.4**	**21.6**	**21.6.1**	**21.6.3**	**21.6.4**	**99**
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates **Event**	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
Protocol error or Internal error *(Concluded 3 of 3)*	^Danyb, Nch [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [CRNALL] [OWEDONE] [RBNEXTSB] 21.5	^Danyb, Nch ^D2pc [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [VNrpdcrF] [VNrpdhrF] [CRNALL] [OWEDONE] [RBNEXTSB] [COUNTCOM] 21.5.4		[SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [CRNALL] [OWEDONE] [RBNEXTSB] [COUNTCOM] 21.5.4	^Danyb, Nch [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [OWEDONE] [RBNEXTSB] 21.5.4		^Danyb [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [VNrpdcrF] [VNrpdhrF] [CRNALL] [OWEDONE] [COUNTCOM] 21.5.4	^Danyb [SETDIAGTP] [TPABi] [ABDET] [SETDIAG] [AABrqPa] [CRNALL] [OWEDONE] [COUNTCOM] 21.5.4		99
	Db, ^Dchat 21.5				Db 21.5.4					
C-BEGIN ind				[COUNTCOM] [CPSAP] [CBEAFTCO] 21.5.4				[COUNTCOM] [CPSAP] [CBEAFTCO] [CANCEL] 21.6.4		
TP-DONE (heuristic-report) req *(Continued on sheet 6 of 21)*	Dd, ^Dfdone Ncmtr [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5	Dd, ^Dfdone Ncmtr [LOGDAM] [VDcrpaF] [COUNTREPDO] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.1	Dd, ^Dfdone Dhrsfu, Ncmtr [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.2	Dd, ^Dfdone Dhrsfu, Ncmtr [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.3	Dd, ^Dfdone Dhrsfu, ^Db, Ncmtr [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.4	Dd, ^Dfdone Ncmtr [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.6	Dd, ^Dfdone Ncmtr [LOGDAM] [VDcrpaF] [COUNTREPDO] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.4	Dd, ^Dfdone Dhrsfu, Ncmtr [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.3	Dd, ^Dfdone Dhrsfu, Ncmtr [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.6.4	^Dcoor, Dsup Dd, ^Dfdone Ncmtr [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 99

Table A.15-2/X.862 – Commitment (sheet 6 of 21)

State	21.5	21.5.1	21.5.2	21.5.3	21.5.4	21.6	21.6.1	21.6.3	21.6.4	99
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates Event	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
TP-DONE (heuristic-report) req *(Concluded 2 of 2)*					Dd, ^Dfdone Dhrsfu, Db, Ncmtr [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.4					^Dcoor, ^Dsup Dd, ^Dfdone Ncmtr [LOGDAM] [VDcrpaF] [COUNTREPDO] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 99
TP-DONE(heuristic-report, completion-report) req	Dd, ^Dfdone Ncmtr Dcrpa [LOGDAM] [SAVECR] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5	Dd, ^Dfdone Ncmtr Dcrpa [LOGDAM] [SAVECR] [VDcrpaF] [COUNTREPDO] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.1	Dd, ^Dfdone Dhrsfu, Ncmtr Dcrpa [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.2	Dd, ^Dfdone Dhrsfu, Ncmtr Dcrpa [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.3	Dd, ^Dfdone Dhrsfu, Ncmtr Dcrpa [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.4	Dd, ^Dfdone Ncmtr Dcrpa [LOGDAM] [SAVECR] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.6	Dd, ^Dfdone Ncmtr Dcrpa [LOGDAM] [SAVECR] [VDcrpaF] [COUNTREPDO] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.4	Dd, ^Dfdone Dhrsfu, Ncmtr Dcrpa [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.3	Dd, ^Dfdone Dhrsfu, Ncmtr Dcrpa [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.6.4	^Dcoor, Dsup Dd, ^Dfdone Ncmtr Dcrpa [LOGDAM] [SAVECR] [VDcrpaF] [COUNTREPDO] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 99
										^Dcoor, ^Dsup Dd, ^Dfdone Ncmtr Dcrpa [LOGDAM] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 99

Table A.15-2/X.862 – Commitment (sheet 7 of 21)

State	**21.5**	**21.5.1**	**21.5.2**	**21.5.3**	**21.5.4**	**21.6**	**21.6.1**	**21.6.3**	**21.6.4**	**99**
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates **Event**	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
TP-DONE (completion-report) req	Dd Dcrpa [SAVECR] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5	Dd Dcrpa [SAVECR] [VDcrpaF] [COUNTREPDO] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.1	Dd Dcrpa [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.2	Dd Dcrpa [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.3	Dd Dcrpa [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.4	Dd Dcrpa [SAVECR] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.6	Dd Dcrpa [SAVECR] [VDcrpaF] [COUNTREPDO] [VDfdoneT] [VNfaF] [VDdF] 21.6.1	Dd Dcrpa [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.6.2	Dd Dcrpa [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.6.4	^Dcoor, Dsup Dd Dcrpa [SAVECR] [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 99
										^Dcoor, ^Dsup Dd Dcrpa [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 99
TP-DONE req	Dd [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5	Dd [VDcrpaF] [COUNTREPDO] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.1	Dd [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.2	Dd [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.3	Dd [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.5.4	Dd [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.6	Dd [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.6.1	Dd [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.6.3	Dd [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 21.6.4	^Dcoor, Dsup Dd [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 99
										^Dcoor, ^Dsup Dd [VDcrpaF] [COUNTCOM] [VDfdoneT] [VNfaF] [VDdF] 99

Table A.15-2/X.862 – Commitment (sheet 8 of 21)

State	**21.5**	**21.5.1**	**21.5.2**	**21.5.3**	**21.5.4**	**21.6**	**21.6.1**	**21.6.3**	**21.6.4**	**99**
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates **Event**	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
C-COMMIT cnf			^Danyb, ^De [COUNTCOM] [CPSAP] 21.5.4							
			^Danyb, De [SDETrqF] [ABDET] [COUNTCOM] 21.5.4							
			^Dbpart, Dtb [AABrqUd] [SDETrqF] [ABDET] [COUNTCOM] 21.5.4							
			Dbpart [SDETrqF] [ABDET] [COUNTCOM] 21.5.4							
AF-ABORT (user, commitRC) ind			^Danyb [TUABi] [SDETrqF] [ABDET] [OWEDONE] [COUNTCOM] 21.5.4							
			Danyb ^Dbpart [SDETrqF] [ABDET] [COUNTCOM] 21.5.4							
C-CANCEL ind	^Du, Dcancfu [VDcancrT] [RBNEXTSB] 21.5	^Du, Dcancfu [VDcancrT] [RBNEXTSB] 21.5.1		^Du, Dcancfu [VDcancrT] [RBNEXTSB] 21.5.3		21.6	21.6.1	21.6.3		

Table A.15-2/X.862 – Commitment (sheet 9 of 21)

State	21.5	21.5.1	21.5.2	21.5.3	21.5.4	21.6	21.6.1	21.6.3	21.6.4	99
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates Event	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
CAF-RECOVER (ready) ind		^Danyb [RECVRCOMR] [DIALOGUE] [VDchatT] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [VNrpdcrF] [CRNALL] [OWEDONE] 21.5.1	^Danyb ^Dhrsfu [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [NEWCHANNEL] [VDchatT] [ABDET] [VNrpdcrF] [SENDREP?] [CRNALL] [OWEDONE] 21.5.2		^D2pc Db [CRErsU] [CAFDETrqF] 21.5.4					Dcoor, Dsup [CRErsU] [CAFDETrqF] [INITRB] [SETDIAGLO] [TRBi] [OWEDONE] 23.8
			^Danyb Dhrsfu [VDchatT] [RECVRCOMR] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [VNrpdcrF] [CRNALL] [OWEDONE] 21.5.2		D2pc Db [CRErsRTSP] [CAFDETrqF] 21.5.4					Dcoor, ^Dsup [CRErsU] [CAFDETrqF] [INITRB] [SETDIAGLO] [TRBi] [OWEDONE] [COUNTRB] 23.2
										^Dcoor, ^Dsup [RECVRCOMR] [VDchatT] [VDrvypF] 21.1
										^Dcoor, Dsup Dhrsfu [RECVRCOMR] [VDchatT] [VDrvypF] 21.5.2

(Continued on sheet 10 of 21)

Table A.15-2/X.862 – Commitment (sheet 10 of 21)

State	**21.5**	**21.5.1**	**21.5.2**	**21.5.3**	**21.5.4**	**21.6**	**21.6.1**	**21.6.3**	**21.6.4**	**99**
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates **Event**	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
CAF-RECOVER (ready) ind *(Concluded 2 of 2)*		Danyb, ^Db [RECVRCOMR] [DIALOGUE] [VDchatT] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [VNrpdcrF] 21.5.1	Danyb, ^Db ^Dhrsfu [DIALOGUE] [VDchatT] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [VNrpdcrF] [SENDREP?] 21.5.2							^Dcoor, Dsup ^Dhrsfu, Nhrst, Dfdone [VDchatT] [SENDREP?] [VDrvypF] 21.5.2
		Db, ^Dchat [RECVRCOMR] [VDchatT] 21.5.1	Danyb, ^Db Dhrsfu [RECVRCOMR] [VDchatT] [DIALOGUE] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [VNrpdcrF] 21.5.2							^Dcoor, Dsup ^Dhrsfu, Nhrst, ^Dfdone [RECVRCOMR] [VDchatT] [VDrvypF] 21.5.1
		Dchat [CRErsRTSP] [CAFDETrqF] 21.5.1	Dchat [CRErsRTSP] [CAFDETrqF] 21.5.2							^Dcoor, Dsup ^Dhrsfu, ^Nhrst [RECVRCOMR] [VDchatT] [VDrvypF] 21.5.1

Table A.15-2/X.862 – Commitment (sheet 11 of 21)

State	21.5	21.5.1	21.5.2	21.5.3	21.5.4	21.6	21.6.1	21.6.3	21.6.4	99
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates Event	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
CAF-RECOVER (commit) ind	Nch, ^Danyb Ldretry [CRErsRTSP] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [RBNEXTSB] [OWEDONE] 21.5					^Danyb Ldretry [CRErsRTSP] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [OWEDONE] 21.5				^Dsup, Dcoor Dhrsfu, ^Ldretry [VDchatT] [VDrvypF] [RECCOM] [OWEDONECO] 21.3
										^Dsup, Dcoor Dhrsfu, Ldretry [VDrvypF] [CRErsRTSP] [CAFDETrqF] [RECCOM] [OWEDONECO] 21.3
	Nch, ^Danyb ^Ldretry [VDchatT] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [RBNEXTSB] [OWEDONE] 21.5					^Danyb ^Ldretry [VDchatT] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] [OWEDONE] 21.5				^Dsup, Dcoor ^Dhrsfu [VDrvypF] [CRErsRTSP] [CAFDETrqF] [RECCOM] [OWEDONECO] 21.1
	^Nch, ^Danyb Ldretry [CRErsRTSP] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [OWEDONE] 21.5									Dsup Dcoor ^Ldretry [VDchatT] [VDrvypF] [RECCOM] [OWEDONECO] 21.5

(Continued on sheet 12 of 21)

Table A.15-2/X.862 – Commitment (sheet 12 of 21)

State	**21.5**	**21.5.1**	**21.5.2**	**21.5.3**	**21.5.4**	**21.6**	**21.6.1**	**21.6.3**	**21.6.4**	**99**
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates **Event**	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
CAF-RECOVER (commit) ind *(Continued 2 of 3)*	^Nch, ^Danyb ^Ldretry [DIALOGUE] [VDchatT] [SETDIAGTP] [TPABi] [SETDIAG] [NEWCHANNEL] [AABrqPa] [ABDET] [OWEDONE] 21.5									Dsup Dcoor Ldretry [VDrvypF] [CRErsRTSP] [CAFDETrqF] [RECCOM] [OWEDONECO] 21.5
	Danyb, ^Db ^Ldretry [DIALOGUE] [VDchatT] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] 21.5					Danyb, ^Db ^Ldretry [DIALOGUE] [VDchatT] [SETDIAG] [AABrqPa] [NEWCHANNEL] [ABDET] 21.5				
	Danyb, ^Db Ldretry [CRErsRTSP] [CAFDETrqF] [DIALOGUE] [SETDIAG] [AABrqPa] [ABDET] 21.5					Danyb, ^Db Ldretry [CRErsRTSP] [CAFDETrqF] [DIALOGUE] [SETDIAG] [AABrqPa] [ABDET] 21.5				
(Continued on sheet 13 of 21)	Danyb, Db ^Dchat Ldretry [CRErsRTSP] [CAFDETrqF] 21.5									

Table A.15-2/X.862 – Commitment (sheet 13 of 21)

State	**21.5**	**21.5.1**	**21.5.2**	**21.5.3**	**21.5.4**	**21.6**	**21.6.1**	**21.6.3**	**21.6.4**	**99**
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates **Event**	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
CAF-RECOVER (commit) ind *(Concluded 3 of 3)*	Danyb, Db ^Dchat ^Ldretry [VDchatT] 21.5									
	Danyb, Db Dchat ^Ldretry Ldretryo [OLDCHANNEL] [CRErsRTSP] [CAFDETrqF] [NEWCHANNEL] 21.5									
	Danyb, Db Dchat Ldretry ^Ldretryo [CRErsRTSP] [CAFDETrqF] [NEWCHANNEL] 21.5									
	Danyb, Db Dchat Ldretry Ldretryo [CRErsRTSP] [CAFDETrqF] [OLDCHANNEL] [CRErsRTSP] [CAFDETrqF] [VDchatF] 21.5									

Table A.15-2/X.862 – Commitment (sheet 14 of 21)

State	21.5	21.5.1	21.5.2	21.5.3	21.5.4	21.6	21.6.1	21.6.3	21.6.4	99
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates Event	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
CAF-RECOVER (commit, heuristic-report) ind										^Dsup, Dcoor ^Dhrsfu Ldretry [CRErsRTSB] [CAFDETrqF] [VDrvypF] [RECCOM] [TREP] [LOGDAM] [OWEDONECO] [COUNTREP] [COUNTCOM] 21.3
										^Dsup, Dcoor ^Dhrsfu ^Ldretry [VDchatT] [VDrvypF] [RECCOM] [TREP] [LOGDAM] [OWEDONECO] [COUNTREP] [COUNTCOM] 21.3
C-RECOVER (done) cnf			^Atwr [CAFDETrqF] [VDchatF] [COUNTCOM] 21.5.4							
			Atwr, ^Atokx [CAFDETrqF] [VDchatF] [COUNTCOM] 21.5.4							

Table A.15-2/X.862 – Commitment (sheet 15 of 21)

State	21.5	21.5.1	21.5.2	21.5.3	21.5.4	21.6	21.6.1	21.6.3	21.6.4	99
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates **Event**	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
C-RECOVER (retry-later) cnf		^Dhrsfu, ^Atwr [CAFDETrqF] [VDchatF] 21.5.1	^Atwr [CAFDETrqF] [VDchatF] [VDrvypT] 99							
		^Dhrsfu, Atwr, ^Atokx [CAFDETrqF] [VDchatF] 21.5.1	Atwr, ^Atokx [CAFDETrqF] [VDchatF] [VDrvypT] 99							
AF-TOKEN-GIVE (two-way-recovery) ind	Atwr, Atokx Dchat [VAtokxF] 21.5		Atwr, Atokx Dchat [VAtokxF] 21.5.2							
AF-TOKEN-PLEASE ind	Dchat 21.5		Dchat 21.5.2							
CAF-GIVE ind	[CAFDETrqNU] 21.5	D2pc [CAFDETrqNU] 21.5.1	D2pc [CAFDETrqNU] 21.5.2		D2pc [CAFDETrqNU] 21.5.4					Dcoor, ^Drvyp [VDchatT] [RECVRRDY] 20.3
										^Dcoor, ^Dsup, ^Drvyp [VDchatT] [RECVRCOMI] 21.1
										^Dcoor, Dsup ^Dhrsfu, Nhrst, Dfdone [VDrvypF] [VDchatT] [SENDREP?] 21.5.2
										^Dcoor, Dsup ^Dhrsfu, Nhrst, ^Dfdone [RECVRCOMI] [VDchatT] [VDrvypF] 21.5.1

(Continued on sheet 16 of 21)

Table A.15-2/X.862 – Commitment (sheet 16 of 21)

State	**21.5**	**21.5.1**	**21.5.2**	**21.5.3**	**21.5.4**	**21.6**	**21.6.1**	**21.6.3**	**21.6.4**	**99**
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates **Event**	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
CAF-GIVE ind *(Concluded 2 of 2)*										^Dcoor, Dsup ^Dhrsfu, ^Nhrst [RECVRCOMI] [VDchatT] [VDrvypF] 21.5.1
										^Dcoor, Dsup Dhrsfu [RECVRCOMI] [VDchatT] [VDrvypF] 21.5.2
CAF-FAIL ind	21.5		21.5.2		D2pc 21.5.4					Dsup, ^Drvyp [CAFPLrqSP] 99
										^Dsup, ^Drvyp [CAFPLrqSB] 99
Heuristic-decision										Dsup [LOGHD] 99
										^Dsup 99
Heuristic-damage-comp	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 21.5	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 21.5.1	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 21.5.2		memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 21.5.4	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 21.6				memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 99
Delay-recovery	Dchat [CRErsRTSP] [CAFDETrqF] [VdchatF] 21.5									
Retry-recovery										Drvyp, Dsup [CAFPLrqSP] [VDrvypF] 99
										Drvyp, ^Dsup [CAFPLrqSB] [VDrvypF] 99

Table A.15-2/X.862 – Commitment (sheet 17 of 21)

State	21.5	21.5.1	21.5.2	21.5.3	21.5.4	21.6	21.6.1	21.6.3	21.6.4	99
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates Event	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
Restart-TPPM	21.5	^Dhrsfu [INITREPSP] [CAFPLrqSP] 99	Dhrsfu [CAFPLrqSP] 99							
Rollback-all Enter-ready										
cr-allowed	[VDcrpaT] 21.5	[VDcrpaT] 21.5.1	[VDcrpaT] 21.5.2	[VDcrpaT] 21.5.3	[VDcrpaT] 21.5.4	[VDcrpaT] 21.6	[VDcrpaT] 21.6.1	[VDcrpaT] 21.6.2	[VDcrpaT] 21.6.3	[VDcrpaT] 99
cr-not-allowed	[VDcrpaF] 21.5	[VDcrpaF] 21.5.1	[VDcrpaF] 21.5.2	[VDcrpaF] 21.5.3	[VDcrpaF] 21.5.4	[VDcrpaF] 21.6	[VDcrpaF] 21.6.1	[VDcrpaF] 21.6.2	[VDcrpaF] 21.6.3	[VDcrpaF] 99
Set-done-true	[VDdT] 21.5	[VDdT] 21.5.1	[VDdT] 21.5.2	[VDdT] 21.5.3	[VDdT] 21.5.4	[VDdT] 21.6	[VDdT] 21.6.1	[VDdT] 21.6.3	[VDdT] 21.6.4	[VDdT] 99
Continue-commit	21.5				21.5.4					
Continue-unknown report-status		21.5.1		21.5.3	21.5.4		21.6.1			Dsup 99
										^Dsup, Dhrsfu 99
										^Dsup, ^Dhrsfu ^Dnchra 99
										^Dsup, ^Dhrsfu Dnchra [VNhrstF] [VNcnthrINC] 99
rewrite-log	Dcoor, D2pc [REWRLOG] 21.5									Dcoor, D2pc [REWRLOG] 99
log-rewritten	Dhrsfu, ^Dcdfu [EARLYC] 21.5	21.5.1	21.5.2	21.5.3	21.5.4	Dhrsfu, ^Dcdfu [EARLYC] 21.6	21.6.1	21.6.3	21.6.4	99
	Dhrsfu, Dcdfu 21.5					Dhrsfu, Dcdfu 21.6				
	^Dhrsfu 21.5					^Dhrsfu 21.6				

Table A.15-2/X.862 – Commitment (sheet 18 of 21)

State	**21.5**	**21.5.1**	**21.5.2**	**21.5.3**	**21.5.4**	**21.6**	**21.6.1**	**21.6.3**	**21.6.4**	**99**
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates **Event**	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
send-report		D2pc, ^Danyb [SENDREP?] 21.5.2					^D2pc, ^Danyb [SENDREP?] 21.6.3			[CAFPLrqSP] [VDrvypF] 99
		D2pc, Dtb [SENDREP?AB] 21.5.2					Danyb [SENDREP?] 21.6.3			
		D2pc, Db, ^Dchat [CAFPLrqSP] 99								
		D2pc, Db, Dchat [CRErsRTSP] [CAFDETrqCU] [CAFPLrqSP] [VDchatF] 99								
		^D2pc, ^Danyb, Nch [SENDREP?] 21.5.3								
		^D2pc, ^Danyb, ^Nch [SENDREP?] [CPSAP] 21.5.4								
		^D2pc, Danu [SENDREP?AB] [AABrqUd] [SDETrqF] [ABDET] [COUNTCOM] 21.5.4								

Table A.15-2/X.862 – Commitment (sheet 19 of 21)

State	21.5	21.5.1	21.5.2	21.5.3	21.5.4	21.6	21.6.1	21.6.3	21.6.4	99
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates Event	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
Rollback-next-trans	^Danu, ^Db, Nch 21.6	^D2pc,^Danu, ^Db, Nch 21.6.1	21.5.2	^Db 21.6.3	^D2pc,^Db, Nch 21.6.4					99
	^Danu, Db, Nch 21.5									
	Danu, ^Db, Nch 21.5	^D2pc, Danu, ^Db 21.5.1								
	Danu, Db, Nch 21.5				Db 21.5.4					
	^Nch 21.5	^Nch 21.5.1			^Nch, ^Db 21.5.4					
Complete-commit	Danyb, ^Db Danu [COMRSP] [CRBrq] [DELBR] [SDETrqRBC] [CMPCOM] 1					[COMRSP] [CMPCOMSP] [CMPCOM] [CANCEL] 23.3			[CMPCOMSP] [CMPCOM] [CANCEL] 23.3	
	Danyb, ^Db ^Danu, ^Nrpend [COMRSP] [DELBR] [SDETrqF] [CMPCOM] 1				Danyb, ^Db ^Danu, ^Nrpend [DELBR] [SDETrqF] [CMPCOM] 1					
	Danyb, ^Db ^Danu, Nrpend [COMRSP] [SDETrqF] [ABDET] [CMPCOM] 23.8				Danyb, ^Db ^Danu, Nrpend [SDETrqF] [ABDET] [CMPCOM] 23.8					
(Continued on sheet 20 of 21)	Danyb, Db ^Dchat ^Nrpend [DELBR] [CMPCOM] 1				Danyb, Db ^Nrpend [DELBR] [CMPCOM] 1					

Table A.15-2/X.862 – Commitment (sheet 20 of 21)

State	21.5	21.5.1	21.5.2	21.5.3	21.5.4	21.6	21.6.1	21.6.3	21.6.4	99
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates Event	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
Complete-commit *(Continued 2 of 3)*	Danyb, Db ^Dchat Nrpend [CMPCOM] 23.8				Danyb, Db Nrpend [CMPCOM] 23.8					
	Danyb, Db Dchat ^Nrpend [RECVRDONE] [DELBR] [CAFDETrqF] [VDchatF] [CMPCOM] 1									
	Danyb, Db Dchat Nrpend [RECVRDONE] [CAFDETrqF] [VDchatF] [CMPCOM] 23.8									
	^Danyb, De [COMRSP] [DELBR] [SDETrqF] [CMPCOM] 1				^Danyb, De [DELBR] [SDETrqF] [CMPCOM] 1					
(Continued on sheet 21 of 21)	^Danyb, ^De Nch, Dsh [COMRSP] [CMPCOMSP] [RESETD] [CMPCOM] 2				^Danyb, ^De Nch, Dsh [OPSAP] [CMPCOMSP] [RESETD] [CMPCOM] 2					

Table A.15-2/X.862 – Commitment (sheet 21 of 21)

State	**21.5**	**21.5.1**	**21.5.2**	**21.5.3**	**21.5.4**	**21.6**	**21.6.1**	**21.6.3**	**21.6.4**	**99**
Node state	DECIDED (commit) or DECIDED (unknown)									
	Base standard 2PC decision received from superior confirming awaited	2PC / OP reporting to superior (dialogue or channel) awaited	2PC cmt cnf from superior awaited	OP / RO / EE C-BEGIN ind awaited	No 1992 dialogue all done, psap closed local complete awaited	Decision received from superior confirming awaited need to rollback	OP reporting to superior on data awaited need to rollback	OP C-BEGIN ind awaited need to rollback	psap closed local complete awaited need to rollback	Channel establishment awaited
Predicates **Event**	Dl, Dsup, Dcoor D2pc	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu)	^Dcoor, Dl, Dsup, D2pc, ^Nch	Dl, Dsup, ^D2pc, Nch ^Danyb	Dl, Dsup	Dcoor, Dl, Dsup, Nch, Nrpend	^Dcoor, Dl, Dsup, (^Dhrsfu or Dcdfu), Nch, Nrpend	^Dcoor, Dl, Dsup, Nch Nrpend, ^Do	^Dcoor, Dl, Dsup, Nch, Nrpend	Dl ^Dchat
Complete-commit *(Concluded 3 of 3)*	^Danyb, ^De Nch, ^Dsh, Dg [COMRSP] [CMPCOMSP] [RESETD] [CMPCOM] 2				^Danyb, ^De Nch, ^Dsh, Dg [OPSAP] [CMPCOMSP] [RESETD] [CMPCOM] 2					
	^Danyb, ^De Nch, ^Dsh, ^Dg [COMRSP] [CMPCOMSP] [RESETD] [CMPCOM] 3				^Danyb, ^De Nch, ^Dsh, ^Dg [OPSAP] [CMPCOMSP] [RESETD] [CMPCOM] 3					
	^Danyb, ^De ^Nch, Dsh [COMRSP] [CMPCOMSP] [DELBR] [RESETD] [CMPCOM] 2				^Danyb, ^De ^Nch, Dsh [OPSAP] [CMPCOMSP] [DELBR] [RESETD] [CMPCOM] 2					
	^Danyb, ^De ^Nch, ^Dsh, Dg [COMRSP] [CMPCOMSP] [DELBR] [RESETD] [CMPCOM] 2				^Danyb, ^De ^Nch, ^Dsh, Dg [OPSAP] [CMPCOMSP] [DELBR] [RESETD] [CMPCOM] 2					
	^Danyb, ^De ^Nch, ^Dsh, ^Dg [COMRSP] [CMPCOMSP] [DELBR] [RESETD] [CMPCOM] 3				^Danyb, ^De ^Nch, ^Dsh, ^Dg [OPSAP] [CMPCOMSP] [DELBR] [RESETD] [CMPCOM] 3					

End of Table A.15-2/X.862

Table A.15-3/X.862 – Commitment (sheet 1 of 2)

State	25	26.1	26.2	26.3	26.4
Node state		No transaction	active/ready	commit	rollback
	zombie stillborn transaction branch	NFSM dormant	NFSM awake	NFSM awake termination phase	NFSM awake termination phase
Predicates Event	Dl	Dl	Dl	Dl	Dl
Protocol error or Internal error	25	26.1	26.2	26.3	26.4
TP-COMMIT req	[COUNTRDY] [VNcmtrT] [COUNTGE] [VNtT] 25		[COUNTRDY] [VNcmtrT] [COUNTGE] [VNtT] 26.2		
TP-ONE-PHASE req	[COUNTRDY] [COUNTGE] [VNtT] 25		[COUNTRDY] [VNoprT] [COUNTGE] [VNtT] 26.2		
TP-READ-ONLY req	[COUNTRDY] [COUNTGE] [VNtT] 25		^Nr [COUNTRDY] [VNrorT] [COUNTGE] [VNtT] 26.2		
TP-EARLY-EXIT req	[COUNTRDY] [COUNTGE] [VNtT] 25		^Nr [COUNTRDY] [VNeerT] [COUNTGE] [VNtT] 26.2		
TP-ROLLBACK req	^Nfrb [INITRB] [OWEDONE] [COUNTRB] 23.2		^Nfrb [INITRB] [OWEDONE] 26.4		
	Nfrb [COUNTRB] 23.2		Nfrb 26.4		
TP-DONE (heuristic-report) req				^Nror, ^Nopr, ^Neer, Dd, ^Dfdone [LOGDAM] [VDfdoneT] [VNfaF] [VDdF] [COUNTCOM] 26.3	^Nror, ^Nopr, ^Neer, Dd, ^Dfdone [LOGDAM] [VDfdoneT] [VNfaF] [VDdF] [COUNTRB] 26.4
TP-DONE req				Dd [VNfaF] [VDdF] [COUNTCOM] 26.3	Dd, ^Drbrep [VNfaF] [VDdF] [COUNTRB] 26.4
					Dd, Drbrep [VNfaF] [VDdF] 26.4
Heuristic-decision	25		26.2	26.3	
Heuristic-damage-comp	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 25				
Rollback-by-TPPM	^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] [COUNTRB] 23.2		^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] 26.4	^Nfrb [SETDIAGLO] [TRBi] [INITRB] [OWEDONE] 26.4	
	Nfrb [COUNTRB] 23.2		Nfrb 26.4	Nfrb 26.4	

Table A.15-3/X.862 – Commitment (sheet 2 of 2)

State	**25**	**26.1**	**26.2**	**26.3**	**26.4**
Node state		No transaction	active/ready	commit	rollback
	zombie stillborn transaction branch	NFSM dormant	NFSM awake	NFSM awake termination phase	NFSM awake termination phase
Predicates **Event**	Dl	Dl	Dl	Dl	Dl
Rollback-all	[COUNTRB] 23.2		26.4		26.4
complete-rollback					[CMPRB] [VDrbrepF] 26.4
report-rollback					[VDrbrepT] 26.4
send-prepare or send-ready? or one-ready or Enter-ready-state or enter-one-phase-state or enter-read-only-state or enter-early-exit-state or rollback-next-ta	25		26.2	26.3	26.4
report-status or send-report or log-rewritten				26.3	
cr-allowed cr-not-allowed				26.3	26.4
Set-done-true	[VDdT] 25			[VDdT] 26.3	[VDdT] 26.4
Continue-commit or continue-unknown	21.3		26.3		
complete-commit				^Nrpend [CMPCOM] 26.3 Nrpend [CMPCOM] 26.4	
activate-nfsm	25	26.2		26.2	26.2
deactivate-nfsm	25		[VNcntgeDEC] [VNcntcDEC] 26.1	26.1	26.1

End of Table A.15-3/X.862

Table A.16/X.862 – Rollback (sheet 1 of 24)

State	23.1	23.2	23.3	23.4	23.5	23.6	23.7	23.8
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates **Event**	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
TP-BEGIN-DIALOGUE (accepted) rsp				Nlf Ncr [VNcrF] [VDahT] 23.4				
TP-BEGIN-DIALOGUE (rejected) rsp				^Nrn, ^Da [ABDrsRUrbc] [REJTRAN] [SDETrqF] 1				
AF-BEGIN-DIALOGUE (accepted, dataRI) cnf	^Danyb Dcr [TBDcX] [VDcrF] [VDaT] 23.1							
	Danyb Dcr [VDcrF] [VDaT] 23.1							
	^Dcr [VDaT] 23.1							
AF-BEGIN-DIALOGUE (accepted, rollbackRC) cnf	^Du, ^Dtb Dcr [TBDcX] [VDcrF] [VDaT] [COUNTRB] 23.2							
	Du, ^Dtb Dcr [TBDcX] [VDcrF] [VDaT] [CPSAP] [COUNTRB] 23.2							

(Continued on sheet 2 of 24)

Table A.16/X.862 – Rollback (sheet 2 of 24)

State	**23.1**	**23.2**	**23.3**	**23.4**	**23.5**	**23.6**	**23.7**	**23.8**
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates **Event**	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
AF-BEGIN-DIALOGUE (accepted, rollbackRC) cnf *(Concluded 2 of 2)*	Dtb [AABrqUd] [SDETrqF] [ABDET] [COUNTRB] 23.2							
AF-BEGIN-DIALOGUE (rejected(provider), dataRI) cnf	^Danyb [TBDcX] [SDETrqRBC] [ABDET] [NOTCHAIN] [OWEDONE] [COUNTRB] 23.2							
	Danyb [SDETrqRBC] [ABDET] [NOTCHAIN] [COUNTRB] 23.2							
AF-BEGIN-DIALOGUE (rejected(user), dataRI) cnf	^Danyb, Du [TBDcX] [SDETrqRBC] [ABDET] [OWEDONE] [COUNTRB] 23.2							
	Danyb, Du [SDETrqRBC] [ABDET] [COUNTRB] 23.2							
AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) cnf *(Continued on sheet 3 of 24)*	^Danyb [TBDcX] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [OWEDONE] [COUNTRB] 23.2							

Table A.16/X.862 – Rollback (sheet 3 of 24)

State	23.1	23.2	23.3	23.4	23.5	23.6	23.7	23.8
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates Event	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) cnf *(Concluded 2 of 2)*	Danyb [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [COUNTRB] 23.2							
AF-BEGIN-DIALOGUE (rejected(user), rollbackRC) cnf	^Danyb [TBDcX] [SDETrqF] [ABDET] [NOTCHAIN] [OWEDONE] [COUNTRB] 23.2							
	Danyb [SDETrqF] [ABDET] [NOTCHAIN] [COUNTRB] 23.2							
SAF-ASSOCIATION-LOST ind	^Danyb [TBDcRP] [ABDET] [NOTCHAIN] [OWEDONE] [COUNTRB] 23.2							
	Danyb [ABDET] [NOTCHAIN] [COUNTRB] 23.2							

Table A.16/X.862 – Rollback (sheet 4 of 24)

State	23.1	23.2	23.3	23.4	23.5	23.6	23.7	23.8
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates Event	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
AF-END-DIALOGUE (confirmation = FALSE) ind	Dx, ^Danyb [TPABiBTED] [SDETrqRBC] [ABDET] [OWEDONE] [COUNTRB] 23.2							
	Dx, Danyb [SDETrqRBC] [ABDET] [COUNTRB] 23.2							
AF-END-DIALOGUE (confirmation = TRUE) ind	Dx, ^Danyb Denbb=0 [TPABiBTED] [SDETrqRBCR] [ABDET] [OWEDONE] [COUNTRB] 23.2							
	Dx, Danyb Denbb=0 [SDETrqRBCR] [ABDET] [COUNTRB] 23.2							
	Dx Denbb>0 [DECDENB] 23.1							
AF-U-ERROR ind	23.1		23.3					
AF-U-ERROR cnf	Dsh Denb>0 [DECDENB] 23.1		Dsh Denb>0 [DECDENB] 23.3					

Table A.16/X.862 – Rollback (sheet 5 of 24)

State	**23.1**	**23.2**	**23.3**	**23.4**	**23.5**	**23.6**	**23.7**	**23.8**
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates **Event**	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
TP-U-ABORT req	Nfa, ^Danyb [ABTPSUI] [NOTCHAIN] 23.1	Nfa, ^Danyb [ABDET] [NOTCHAIN] [AABrqUd] [OPSAP] [SDETrqF] 23.2	Nfa, ^Danyb [ABTPSUI] [NOTCHAIN] 23.3	^Ncr, Nfa, ^Danyb [ABTPSUI] [NOTCHAIN] 23.4	Nfa, ^Danyb [ABTPSUI] [NOTCHAIN] 23.5	Nfa [NOTCHAIN] [AABrqUd] [SDETrqBF] [ABDET] 23.7	^Danyb, ^Nch Dd, Nfa [AABrqUd] [OPSAP] [SDETrqBF] [ABDET] 23.7	
							^Danyb, Nch Dd, Nfa [AABrqUr] [OPSAP] [SDETrqRBC] [ABDET] [NOTCHAIN] 23.7	
AF-ABORT (user, dataRI) ind	Dx, ^Danyb [TUABi] [SDETrqRBC] [ABDET] [OWEDONE] [COUNTRB] 23.2	Dch [TUABi] [SDETrqF] [ABDET] [OWEDONE] [NOTCHAIN] 23.2				[TUABi] [SDETrqF] [ABDET] [OWEDONE] [NOTCHAIN] 23.7		
	Dx, Danyb [SDETrqRBC] [ABDET] [COUNTRB] 23.2							
AF-ABORT (provider, abortRI) ind or A-P-ABORT ind or A-ABORT ind or A-ABORT req or A-RELEASE (result = affirmative) rsp or A-RELEASE (result = affirmative) cnf	^Danyb, ^Dps [SETDIAGTP] [TPABi] [ABDET] [OWEDONE] [COUNTRB] [NOTCHAIN] 23.2	^Danyb [SETDIAGTP] [TPABi] [ABDET] [OWEDONE] [NOTCHAIN] 23.2	^Danyb [SETDIAGTP] [TPABi] [CRNALL] [ABDET] [OWEDONE] [NOTCHAIN] 23.8	^Danyb, ^Ncr [SETDIAGTP] [TPABi] [CRNALL] [ABDET] [OWEDONE] [NOTCHAIN] 23.8	^Danyb [SETDIAGTP] [TPABi] [ABDET] [OWEDONE] [NOTCHAIN] 23.7	[SETDIAGTP] [TPABi] [ABDET] [OWEDONE] [NOTCHAIN] 23.7		

(Continued on sheet 6 of 24)

Table A.16/X.862 – Rollback (sheet 6 of 24)

State	23.1	23.2	23.3	23.4	23.5	23.6	23.7	23.8
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates Event	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
AF-ABORT (provider, abortRI) ind or A-P-ABORT ind or A-ABORT ind or A-ABORT req or A-RELEASE (result = affirmative) rsp or A-RELEASE (result = affirmative) cnf *(Concluded 2 of 2)*	Danyb, ^Dps [ABDET] [NOTCHAIN] [COUNTRB] 23.2			^Danyb Ncr [SETDIAGTP] [TPABi] [CRNALL] [ABDET] [REJTRAN] 1 Danyb	Danyb, ^Dd [ABDET] [NOTCHAIN] [NXTTRAN] 23.5 Danyb, Dd			
	^Danyb, Dps [SETDIAGTP] [TPABi] [THRiH] [LOGDAMH] [ABDET] [OWEDONE] [COUNTRB] [NOTCHAIN] 23.2		Danyb [ABDET] [NOTCHAIN] 23.8	[ABDET] [NOTCHAIN] 23.8	[ABDET] [NOTCHAIN] 23.7			
	Danyb, Dps [THRiH] [LOGDAMH] [ABDET] [NOTCHAIN] [COUNTRB] 23.2							
Protocol error or Internal error	^Danyb, ^Dps [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [OWEDONE] [COUNTRB] 23.2	^Danyb [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [OWEDONE] 23.2	^Danyb [SETDIAGTP] [TPABi] [CRNALL] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [OWEDONE] 23.8	^Danyb, ^Ncr [SETDIAGTP] [TPABi] [CRNALL] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [OWEDONE] 23.8	^Danyb [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [OWEDONE] 23.7	[SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [OWEDONE] 23.7	^Danyb [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] 23.7	
(Continued on sheet 7 of 24)	Danyb, ^Dps [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [COUNTRB] 23.2			^Danyb, Ncr [SETDIAGTP] [TPABi] [CRNALL] [SETDIAG] [AABrqPa] [ABDET] [REJTRAN] 1	Danyb, ^Db, ^Dd [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [NXTTRAN] 23.5			

Table A.16/X.862 – Rollback (sheet 7 of 24)

State	**23.1**	**23.2**	**23.3**	**23.4**	**23.5**	**23.6**	**23.7**	**23.8**
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates **Event**	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
Protocol error or Internal error *(Concluded 2 of 2)*		Db 23.2	Danyb, ^Db [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] 23.8	Danyb [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] 23.8	Danyb, ^Db, Dd [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] 23.7		Db 23.7	Db 23.8
	^Danyb, Dps [SETDIAGTP] [TPABi] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [OWEDONE] [COUNTRB] 23.2							
	Danyb, Dps [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [COUNTRB] 23.2							
AF-GRANT-CONTROL ind or AF-REQUEST-CONTROL ind	^Dsh 23.1		^Dsh 23.3					
AF-HANDSHAKE ind or AF-HANDSHAKE cnf	Dh 23.1		Dh 23.3					
AF-HANDSHAKE-AND-GRANT-CONTROL ind or AF-HANDSHAKE-AND-GRANT-CONTROL cnf	Dh, ^Dsh 23.1		Dh, ^Dsh 23.3					

Table A.16/X.862 – Rollback (sheet 8 of 24)

State	**23.1**	**23.2**	**23.3**	**23.4**	**23.5**	**23.6**	**23.7**	**23.8**
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates **Event**	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
C-BEGIN ind						^Dd [NXTTRAN] [CBEAFTRB] 23.6 Dd [CPSAP] [CBEAFTRB] 23.7		
C-BEGIN cnf	[VDbcrT] [VDxF] 23.1							
U-ASE ind	23.1		23.3					
AF-DEFER (end-dialogue) ind			^De 23.3					
AF-DEFER (grant-control) ind			^De, ^Dg 23.3					
AF-PREPARE ind or AF-PREPARE (data-permitted = FALSE) ind or AF-PREPARE (data-permitted = TRUE) ind			23.3					
C-READY ind			Ddyn, Drrec 23.3					
C-NOCHANGE (result-requested) ind			Ddyn, Drrec 23.3 ^Do, Drrec 23.3					
AF-NOCHANGE (result-requested) ind			^Do, Drrec 23.3					
TP-DONE (heuristic-report) req	Dd, ^Dfdone ^Nopr, ^Nror ^Neer ^Nr, Np [VDdF] [VNfaF] [VDfdoneT] [LOGHD] [VDcrpaF] [COUNTRB] 23.1	Dd, ^Dfdone ^Nopr, ^Nror ^Neer ^Nr, Np [VDdF] [VNfaF] [VDfdoneT] [LOGHD] [VDcrpaF] [COUNTRB] 23.2	Dd, ^Dfdone ^Nopr, ^Nror ^Neer Np [VDdF] [VNfaF] [VDfdoneT] [VDaT] [VNrnT] [LOGHD] [VDcrpaF] [COUNTRB] 23.3	Dd, ^Dfdone ^Nopr, ^Nror ^Neer ^Ncr, Np [VDdF] [VNfaF] [VDfdoneT] [VDaT] [VNrnT] [LOGHD] [VDcrpaF] [COUNTRB] 23.4				Dd, ^Dfdone ^Nopr, ^Nror ^Neer Np [VDdF] [VNfaF] [VDfdoneT] [LOGHD] [VDcrpaF] [COUNTRB] 23.8

(Continued on sheet 9 of 24)

Table A.16/X.862 – Rollback (sheet 9 of 24)

State	23.1	23.2	23.3	23.4	23.5	23.6	23.7	23.8
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates Event	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
TP-DONE (heuristic-report) req *(Concluded 2 of 2)*	Dd, ^Dfdone ^Nopr Nr [VDdF] [VNfaF] [VDfdoneT] [LOGHD] [VDcrpaF] [COUNTRB] 23.1	Dd, ^Dfdone ^Nopr Nr [VDdF] [VNfaF] [VDfdoneT] [LOGHD] [VDcrpaF] [COUNTRB] 23.2						
TP-DONE (heuristic-report, completion-report) req	Dcrpa Dd, ^Dfdone ^Nopr, ^Nror ^Neer ^Nr, Np [VDdF] [VNfaF] [VDfdoneT] [LOGHD] [SAVECR] [VDcrpaF] [COUNTRB] 23.1	Dcrpa Dd, ^Dfdone ^Nopr, ^Nror ^Neer ^Nr, Np [VDdF] [VNfaF] [VDfdoneT] [LOGHD] [SAVECR] [VDcrpaF] [COUNTRB] 23.2	Dcrpa Dd, ^Dfdone ^Nopr, ^Nror ^Neer Np [VDdF] [VNfaF] [VDfdoneT] [VDaT] [VNrnT] [LOGHD] [SAVECR] [VDcrpaF] [COUNTRB] 23.3	Dcrpa Dd, ^Dfdone ^Nopr, ^Nror ^Neer ^Ncr, Np [VDdF] [VNfaF] [VDfdoneT] [VDaT] [VNrnT] [LOGHD] [SAVECR] [VDcrpaF] [COUNTRB] 23.4				Dcrpa Dd, ^Dfdone ^Nopr, ^Nror ^Neer ^Ncr, Np [VDdF] [VNfaF] [VDfdoneT] [LOGHD] [SAVECR] [VDcrpaF] [COUNTRB] 23.8
TP-DONE (completion-report) req	Dcrpa Dd [VDdF] [VNfaF] [VDfdoneT] [SAVECR] [VDcrpaF] [COUNTRB] 23.1	Dcrpa Dd [VDdF] [VNfaF] [VDfdoneT] [SAVECR] [VDcrpaF] [COUNTRB] 23.2	Dcrpa Dd [VDdF] [VNfaF] [VDfdoneT] [VDaT] [VNrnT] [SAVECR] [VDcrpaF] [COUNTRB] 23.3	Dcrpa Dd ^Ncr [VDdF] [VNfaF] [VDfdoneT] [VDaT] [VNrnT] [SAVECR] [VDcrpaF] [COUNTRB] 23.4	Dcrpa Ni, Dd [VDdF] [VNfaF] [VDcrpaF] 23.5	Dcrpa Dd [VDdF] [VNfaF] [VDcrpaF] 23.6	Dcrpa Dd [VDdF] [VNfaF] [VDcrpaF] [NXTTRAN] 23.7	Dcrpa Dd, Danyb [VDdF] [VNfaF] [VDfdoneT] [VDcrpaF] [COUNTRB] 23.8

Table A.16/X.862 – Rollback (sheet 10 of 24)

State	**23.1**	**23.2**	**23.3**	**23.4**	**23.5**	**23.6**	**23.7**	**23.8**
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates **Event**	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
TP-DONE req	Dd [VDdF] [VNfaF] [VDfdoneT] [VDcrpaF] [COUNTRB] 23.1	^Drbrep, Dd [VDdF] [VNfaF] [VDfdoneT] [VDcrpaF] [COUNTRB] 23.2	Dd [VDdF] [VNfaF] [VDfdoneT] [VDaT] [VNrnT] [VDcrpaF] [COUNTRB] 23.3	Dd ^Ncr [VDdF] [VNfaF] [VDfdoneT] [VDaT] [VNrnT] [VDcrpaF] [COUNTRB] 23.4	Ni, Dd [VDdF] [VNfaF] [VDcrpaF] 23.5	Dd [VDdF] [VNfaF] [VDcrpaF] 23.6	Dd [VDdF] [VNfaF] [VDcrpaF] [NXTTRAN] 23.7	Dd, Danyb [VDdF] [VNfaF] [VDfdoneT] [VDcrpaF] [COUNTRB] 23.8
		Drbrep, Dd [VDdF] [VNfaF] 23.2						
C-ROLLBACK ind	^Du, ^Danyb [CRBrs] [COUNTRB] 23.2		23.4		^Danyb, Nch [RBRSPNOAB] 23.6			
	Du, ^Danyb [CRBrs] [COUNTRB] [CPSAP] 23.2				^Danyb, ^Nch, ^Dd [RBRSPNOAB] [NXTTRAN] 23.5			
	Danyb [AABrqUrbc] [SDETrqF] [ABDET] [COUNTRB] 23.2				^Danyb, ^Nch, Dd [RBRSPNOAB] [CPSAP] 23.7			
					Danyb, ^Dd [RBRSPAB] [ABDET] [SDETrqF] [NXTTRAN] 23.5			
					Danyb, Dd [RBRSPAB] [SDETrqF] [ABDET] 23.7			

Table A.16/X.862 – Rollback (sheet 11 of 24)

State	23.1	23.2	23.3	23.4	23.5	23.6	23.7	23.8
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates Event	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
AF-EARLY-EXIT ind	^Du, ^Danyb ^Dcdfu [CRBrs] [COUNTRB] 23.2							
	^Du, ^Danyb Dcdfu [SETDIAGEC] [TREP] [CRBrs] [COUNTRB] 23.2							
	Du, ^Danyb ^Dcdfu [CRBrs] [COUNTRB] [CPSAP] 23.2							
	Danyb Dcdfu [SETDIAGEC] [TREP] [AABrqUrbc] [SDETrqF] [ABDET] [COUNTRB] 23.2							
	Danyb ^Dcdfu [AABrqUrbc] [SDETrqF] [ABDET] [COUNTRB] 23.2							
C-CANCEL ind			23.4					

Table A.16/X.862 – Rollback (sheet 12 of 24)

State	23.1	23.2	23.3	23.4	23.5	23.6	23.7	23.8
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates Event	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
AF-REPORT (rollbackRI, heuristic-report) ind	^Du, ^Danyb ^Dhrsfu [TREP] [LOGDAM] [CRBrs] [COUNTRB] 23.2							
	Du, ^Danyb ^Dhrsfu [TREP] [LOGDAM] [CRBrs] [CPSAP] [COUNTRB] 23.2							
	Danyb ^Dhrsfu [TREP] [LOGDAM] [AABrqUrbc] [SDETrqF] [ABDET] [COUNTRB] 23.2							
AF-REPORT (rollbackRI, heuristic-report, completion-report) ind	^Du, ^Danyb ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [CRBrs] [COUNTRB] 23.2							
	Du, ^Danyb ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [CRBrs] [CPSAP] [COUNTRB] 23.2							

(Continued on sheet 13 of 24)

Table A.16/X.862 – Rollback (sheet 13 of 24)

State	23.1	23.2	23.3	23.4	23.5	23.6	23.7	23.8
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates Event	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
AF-REPORT (rollbackRI, heuristic-report, completion-report) ind *(Concluded 2 of 2)*	Danyb ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [AABrqUrbc] [SDETrqF] [ABDET] [COUNTRB] 23.2							
AF-REPORT (rollbackRI, completion-report) ind	^Du, ^Danyb Dcdfu [TREP] [CRBrs] [COUNTRB] 23.2							
	Du, ^Danyb Dcdfu [TREP] [CRBrs] [CPSAP] [COUNTRB] 23.2							
	Danyb Dcdfu [TREP] [AABrqUrbc] [SDETrqF] [ABDET] [COUNTRB] 23.2							
AF-ABORT (provider, diagnostic = begin-transaction-reject, rollbackRI) ind *(Continued on sheet 14 of 24)*	^Danyb [TPABi] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [OWEDONE] [COUNTRB] 23.2							

Table A.16/X.862 – Rollback (sheet 14 of 24)

State	23.1	23.2	23.3	23.4	23.5	23.6	23.7	23.8
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates Event	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
AF-ABORT (provider, diagnostic = begin-transaction-reject, rollbackRI) ind *(Concluded 2 of 2)*	Danyb [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [COUNTRB] 23.2							
AF-ABORT (user, rollbackRI) ind	^Danyb [TUABi] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [OWEDONE] [COUNTRB] 23.2		^Danyb [TUABi] [ABPTNR] [NOTCHAIN] [OWEDONE] 23.4		^Danyb [TUABi] [RBRSPNOAB] [SDETrqF] [ABDET] [NOTCHAIN] [OWEDONE] 23.7			
	Danyb [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [COUNTRB] 23.2		Danyb [ABPTNR] [NOTCHAIN] 23.4		Danyb, Dd [RBRSPNOAB] [SDETrqF] [ABDET] [NOTCHAIN] 23.7			
					Danyb, ^Dd [RBRSPNOAB] [SDETrqF] [ABDET] [NOTCHAIN] [NXTTRAN] 23.5			
AF-ABORT-AND-REPORT (rollbackRI, heuristic-report) ind *(Continued on sheet 15 of 24)*	^Danyb ^Dhrsfu [TREP] [TUABi] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [OWEDONE] [COUNTRB] 23.2							

Table A.16/X.862 – Rollback (sheet 15 of 24)

State	**23.1**	**23.2**	**23.3**	**23.4**	**23.5**	**23.6**	**23.7**	**23.8**
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates **Event**	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
AF-ABORT-AND-REPORT (rollbackRI, heuristic-report) ind *(Concluded 2 of 2)*	Danyb ^Dhrsfu [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [COUNTRB] 23.2							
AF-ABORT-AND-REPORT(rollbackRI, heuristic-report, completion-report) ind	^Danyb ^Dhrsfu, Dcdfu [TREP] [TUABi] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [OWEDONE] [COUNTRB] 23.2							
	Danyb ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [COUNTRB] 23.2							

Table A.16/X.862 – Rollback (sheet 16 of 24)

State	**23.1**	**23.2**	**23.3**	**23.4**	**23.5**	**23.6**	**23.7**	**23.8**
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates **Event**	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
AF-ABORT-AND-REPORT (rollbackRI, completion-report) ind	^Danyb Dcdfu [TUABi] [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [OWEDONE] [COUNTRB] 23.2							
	Danyb Dcdfu [TREP] [CRBrs] [SDETrqF] [ABDET] [NOTCHAIN] [COUNTRB] 23.2							
C-ROLLBACK cnf	^Du, ^Danyb [COUNTRB] 23.2				^Du, ^Danyb 23.6			
					Du, ^Danyb, ^Dd [NXTTRAN] 23.5			
	Du, ^Danyb [COUNTRB] [CPSAP] 23.2				Du, ^Danyb, Dd [CPSAP] 23.7			
	Dtb [AABrqUd] [SDETrqF] [ABDET] [COUNTRB] 23.2				Dtb, ^Dd [AABrqUd] [SDETrqBF] [ABDET] [NXTTRAN] 23.5			
(Continued on sheet 17 of 24)					Dtb, Dd [AABrqUd] [SDETrqBF] [ABDET] 23.7			

Table A.16/X.862 – Rollback (sheet 17 of 24)

State	**23.1**	**23.2**	**23.3**	**23.4**	**23.5**	**23.6**	**23.7**	**23.8**
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates **Event**	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
C-ROLLBACK cnf *(Concluded 2 of 2)*	Dbpart [SDETrqF] [ABDET] [COUNTRB] 23.2				Dbpart, ^Dd [SDETrqF] [ABDET] [NXTTRAN] 23.5			
					Dbpart, Dd [SDETrqF] [ABDET] 23.7			
AF-REPORT (rollbackRC, heuristic-report) ind	^Danyb, Dch ^Dhrsfu [TREP] [LOGDAM] [COUNTRB] 23.2							
	^Danyb, ^Dch ^Dhrsfu [TREP] [LOGDAM] [COUNTRB] [CPSAP] 23.2							
	Dtb ^Dhrsfu [TREP] [LOGDAM] [AABrqUd] [SDETrqF] [ABDET] [COUNTRB] 23.2							
	Dbpart ^Dhrsfu [TREP] [LOGDAM] [SDETrqF] [ABDET] [COUNTRB] 23.2							

Table A.16/X.862 – Rollback (sheet 18 of 24)

State	23.1	23.2	23.3	23.4	23.5	23.6	23.7	23.8
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates Event	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
AF-REPORT (rollbackRC, heuristic-report, completion-report) ind	^Danyb, Dch ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [COUNTRB] 23.2							
	^Danyb, ^Dch ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [COUNTRB] [CPSAP] 23.2							
	Dtb ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [AABrqUd] [SDETrqF] [ABDET] [COUNTRB] 23.2							
	Dbpart ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [SDETrqF] [ABDET] [COUNTRB] 23.2							
AF-REPORT (rollbackRC, completion-report) ind	^Danyb, Dch Dcdfu [TREP] [COUNTRB] 23.2							
(Continued on sheet 19 of 24)	^Danyb, ^Dch Dcdfu [TREP] [COUNTRB] [CPSAP] 23.2							

Table A.16/X.862 – Rollback (sheet 19 of 24)

State	23.1	23.2	23.3	23.4	23.5	23.6	23.7	23.8
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates **Event**	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
AF-REPORT (rollbackRC, completion-report) ind *(Concluded 2 of 2)*	Dtb Dcdfu [TREP] [AABrqUd] [SDETrqF] [ABDET] [COUNTRB] 23.2							
	Dbpart Dcdfu [TREP] [SDETrqF] [ABDET] [COUNTRB] 23.2							
AF-ABORT (provider, diagnostic = begin-transaction-reject, rollbackRC) ind	Du, ^Dbcr ^Danyb [TPABi] [SDETrqF] [ABDET] [OWEDONE] [COUNTRB] 23.2							
	Du, ^Dbcr Danyb [SDETrqF] [ABDET] [COUNTRB] 23.2							
AF-ABORT (user, rollbackRC) ind	^Danyb [TUABi] [SDETrqF] [ABDET] [NOTCHAIN] [OWEDONE] [COUNTRB] 23.2				^Danyb [TUABi] [SDETrqF] [ABDET] [NOTCHAIN] [OWEDONE] 23.7			
(Continued on sheet 20 of 24)	Danyb [SDETrqF] [ABDET] [NOTCHAIN] [COUNTRB] 23.2				Danyb, Dd [SDETrqF] [ABDET] [NOTCHAIN] 23.7			

Table A.16/X.862 – Rollback (sheet 20 of 24)

State	23.1	23.2	23.3	23.4	23.5	23.6	23.7	23.8
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates Event	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
AF-ABORT (user, rollbackRC) ind *(Concluded 2 of 2)*					Danyb, ^Dd [NOTCHAIN] [SDETrqF] [ABDET] [NXTTRAN] 23.5			
AF-ABORT-AND-REPORT (rollbackRC, heuristic-report) ind	^Danyb ^Dhrsfu [TREP] [TUABi] [LOGDAM] [SDETrqF] [ABDET] [NOTCHAIN] [OWEDONE] [COUNTRB] 23.2							
	Danyb ^Dhrsfu [TREP] [LOGDAM] [SDETrqF] [ABDET] [NOTCHAIN] [COUNTRB] 23.2							
AF-ABORT-AND-REPORT(rollbackRC, heuristic-report, completion-report) ind *(Continued on sheet 21 of 24)*	^Danyb ^Dhrsfu, Dcdfu [TREP] [TUABi] [LOGDAM] [SDETrqF] [ABDET] [NOTCHAIN] [OWEDONE] [COUNTRB] 23.2							

Table A.16/X.862 – Rollback (sheet 21 of 24)

State	**23.1**	**23.2**	**23.3**	**23.4**	**23.5**	**23.6**	**23.7**	**23.8**
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates **Event**	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
AF-ABORT-AND-REPORT(rollbackRC, heuristic-report, completion-report) ind *(Concluded 2 of 2)*	Danyb ^Dhrsfu, Dcdfu [TREP] [LOGDAM] [SDETrqF] [ABDET] [NOTCHAIN] [COUNTRB] 23.2							
AF-ABORT-AND-REPORT(rollbackRC, completion-report) ind	^Danyb Dcdfu [TUABi] [TREP] [SDETrqF] [ABDET] [NOTCHAIN] [OWEDONE] [COUNTRB] 23.2							
	Danyb Dcdfu [TREP] [SDETrqF] [ABDET] [NOTCHAIN] [COUNTRB] 23.2							
CAF-RECOVER (ready) ind *(Continued on sheet 22 of 24)*	^Danyb Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [OWEDONE] [COUNTRB] 23.2	^Danyb Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [OWEDONE] 23.2	^Danyb Ddyn, Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [CRNALL] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [OWEDONE] 23.8		^Danyb Ddyn, Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAGTP] [TPABi] [CRNALL] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [OWEDONE] 23.7			Db Ddyn, Drrec [CRErsU] [CAFDETrqF] 23.8

Table A.16/X.862 – Rollback (sheet 22 of 24)

State	23.1	23.2	23.3	23.4	23.5	23.6	23.7	23.8
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates Event	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
CAF-RECOVER (ready) ind *(Concluded 2 of 2)*	Danyb Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [THRiH] [LOGDAMH] [SETDIAG] [AABrqPa] [ABDET] [COUNTRB] 23.2	Db Drrec [CRErsU] [CAFDETrqF] 23.2	Danyb Ddyn, Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAG] [AABrqPa] [ABDET] 23.8		Danyb, ^Dd Ddyn, Drrec [CRErsU] [CAFDETrqF] [DIALOGUE] [SETDIAG] [AABrqPa] [ABDET] [NOTCHAIN] [NXTTRAN] 23.5			
					Danyb, Dd Ddyn, Drrec [ABDET] [NOTCHAIN] 23.7			
CAF-GIVE ind		Db [CAFDETrqNU] 23.2						Db [CAFDETrqNU] 23.8
CAF-FAIL ind		Db 23.2						Db 23.8
Heuristic-damage-comp	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 23.1	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 23.2	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 23.3	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 23.4	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 23.5	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 23.6	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 23.7	memsp (SldD, Naaid, Nbrid) [LOGREMOVE] 23.8
Rollback-all	23.1	23.2	23.3	23.4		23.6	23.7	23.8
cr-allowed	[VDcrpaT] 23.1	[VDcrpaT] 23.2	[VDcrpaT] 23.3	[VDcrpaT] 23.4	[VDcrpaT] 23.5	[VDcrpaT] 23.6	Nfa, Dd [VDcrpaT] 23.7	[VDcrpaT] 23.8
cr-not-allowed	[VDcrpaF] 23.1	[VDcrpaF] 23.2	[VDcrpaF] 23.3	[VDcrpaF] 23.4	[VDcrpaF] 23.5	[VDcrpaF] 23.6	Nfa, Dd [VDcrpaF] 23.7	[VDcrpaF] 23.8
Set-done-true	[VDdT] 23.1	[VDdT] 23.2	[VDdT] 23.3	[VDdT] 23.4	[VDdT] 23.5	[VDdT] 23.6	[VDdT] 23.7	[VDdT] 23.8

Table A.16/X.862 – Rollback (sheet 23 of 24)

State	23.1	23.2	23.3	23.4	23.5	23.6	23.7	23.8
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates Event	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
Report-rollback				^Danyb, Dah ^Nch [ABDrsArbc] [VDahF] [VDaT] [VDrbrepT] [NXTTRAN] 23.4				
				^Danyb, Dah Nch [ABDrsArbc] [VDahF] [VDaT] [VDrbrepT] 23.6				
				^Danyb, ^Dah ^Nch [RBRSPNOAB] [VDaT] [VDrbrepT] [NXTTRAN] 23.4				
				^Danyb, ^Dah Nch [RBRSPNOAB] [VDaT] [VDrbrepT] 23.6				
				Dtb [RBRSPAB] [SDETrqF] [ABDET] [VDaT] [VDrbrepT] [NXTTRAN] 23.8				

(Continued on sheet 24 of 24)

Table A.16/X.862 – Rollback (sheet 24 of 24)

State	**23.1**	**23.2**	**23.3**	**23.4**	**23.5**	**23.6**	**23.7**	**23.8**
	Rollback req issued Rollback confirm awaited	Rollback ind/cnf received Rollback complete awaited	Rollback not received from sup Report to sup awaited	Rollback ind received from sup Report to sup awaited	Rollback req issued Rollback confirm from sup awaited	Report to sup done C-BEGIN awaited	Report to sup done TP-DONE req awaited	Rollback not complete
Predicates **Event**	Dl ^Dsup	Dl ^Dsup	Dl Dsup	Dl, ^Db Dsup	Dl Dsup	Dl, ^Danyb Dsup, Nch	Dl, Dsup	Dl Dsup
Report-rollback *(Concluded 2 of 2)*		[VDrbrepT] 23.2	[RBREQ] [VDaT] [VDrbrepT] 23.5	Dbpart [RBRSPNOAB] [SDETrqF] [ABDET] [VDaT] [VDrbrepT] [NXTTRAN] 23.8				[NXTTRAN] 23.8
Complete-rollback		Danyb [DELBR] [CMPRB] 1			Danyb [DELBR] [CMPRB] 1		Danyb [DELBR] [CMPRB] 1	Danyb [DELBR] [CMPRB] 1
		^Danyb, Dc, ^Du [RESETD] [NXTBR] [CBErq] [CMPRB] 2					^Danyb, Dc, Nch [OPSAP] [RESETD] [CMPRB] 2	
		^Danyb, ^Dc, ^Du [RESETD] [NXTBR] [CBErq] [CMPRB] 3					^Danyb, ^Dc, Nch [OPSAP] [RESETD] [CMPRB] 3	
		^Danyb, Dc, Du [OPSAP] [RESETD] [DELBR] [CMPRB] 2		^Danyb, Dc [OPSAP] [RESETD] [DELBR] [CMPRB] 2	^Danyb, Dc [OPSAP] [RESETD] [DELBR] [CMPRB] 2	Dc [OPSAP] [RESETD] [CMPRB] 2	^Danyb, Dc, ^Nch [OPSAP] [RESETD] [DELBR] [CMPRB] 2	
		^Danyb, ^Dc, Du [OPSAP] [RESETD] [DELBR] [CMPRB] 3		^Danyb, ^Dc [OPSAP] [RESETD] [DELBR] [CMPRB] 3	^Danyb, ^Dc [OPSAP] [RESETD] [DELBR] [CMPRB] 3	^Dc [OPSAP] [RESETD] [CMPRB] 3	^Danyb, ^Dc, ^Nch [OPSAP] [RESETD] [DELBR] [CMPRB] 3	

End of Table A.16 X.862

Table A.17/X.862 – Channel (sheet 1 of 10)

State	1	2	3	4	5	6	7
	Channel does not exist	Free channel, available	Free channel, not available	Channel owned by TPPM	Token awaited CAF-PLEASE req outstanding	AF-BEGIN-DIALOGUE cnf awaited CAF-PLEASE req outstanding	Clean-up
Predicates Event					Atwr		
AF-BEGIN-DIALOGUE (Recovery fu selected, one-way-recovery) ind	^Ldrej [ABDrsAd] [VAtwrF] [VAtppmF] 3						
	Ldrej [SETDIAGBD] [ABDrsRPd] [SDETrqF] 1						
AF-BEGIN-DIALOGUE (Recovery fu selected, two-way-recovery) ind	^Ldrej [ABDrsAd] [VAtwrT] [VAtppmF] 3						
	Ldrej [SETDIAGBD] [ABDrsRPd] [SDETrqF] 1						
AF-BEGIN-DIALOGUE (accepted, dataRI) cnf						Csup memsp(SnD, Caaid, Cbrid) [VAtppmT] [VCinitT] [CAFGIVi] 4	
						^Csup memsb(SnD, Caaid, Cbrid) [VAtppmT] [VCinitT] [CAFGIVi] 4	
						Csup ^memsp(SnD, Caaid, Cbrid) [VCinitT] 2	
						^Csup ^memsb(SnD, Caaid, Cbrid) [VCinitT] 2	

Table A.17/X.862 – Channel (sheet 2 of 10)

State	1	2	3	4	5	6	7
	Channel does not exist	Free channel, available	Free channel, not available	Channel owned by TPPM	Token awaited CAF-PLEASE req outstanding	AF-BEGIN-DIALOGUE cnf awaited CAF-PLEASE req outstanding	Clean-up
Predicates Event					Atwr		
AF-BEGIN-DIALOGUE (rejected(provider), dataRI) cnf						Csup memsp(SnD, Caaid, Cbrid) [CAFFAILi] [SDETrqF] 1	
						^Csup memsb(SnD, Caaid, Cbrid) [CAFFAILi] [SDETrqF] 1	
						Csup ^memsp(SnD, Caaid, Cbrid) [SDETrqF] 1	
						^Csup ^memsb(SnD, Caaid, Cbrid) [SDETrqF] 1	
SAF-ASSOCIATION-LOST ind						Csup memsp(SnD, Caaid, Cbrid) [CAFFAILi] 1	
						^Csup memsb(SnD, Caaid, Cbrid) [CAFFAILi] 1	
						Csup ^memsp(SnD, Caaid, Cbrid) 1	
						^Csup ^memsb(SnD, Caaid, Cbrid) 1	
AF-END-DIALOGUE ind			[SDETrqF] 1		[CAFFAILi] [SDETrqF] 1		

Table A.17/X.862 – Channel (sheet 3 of 10)

State	1	2	3	4	5	6	7
	Channel does not exist	Free channel, available	Free channel, not available	Channel owned by TPPM	Token awaited CAF-PLEASE req outstanding	AF-BEGIN-DIALOGUE cnf awaited CAF-PLEASE req outstanding	Clean-up
Predicates Event					Atwr		
AF-ABORT (provider, abortRI) ind or A-ABORT ind or A-P-ABORT ind or A-RELEASE (result = affirmative) rsp or A-RELEASE (result = affirmative) cnf		1	1		Csup memsp(SnD, Caaid, Cbrid) [CAFFAILi] 1	Csup memsp(SnD, Caaid, Cbrid) [CAFFAILi] 1	1
					^Csup memsb(SnD, Caaid, Cbrid) [CAFFAILi] 1	^Csup memsb(SnD, Caaid, Cbrid) [CAFFAILi] 1	
					Csup ^memsp(SnD, Caaid, Cbrid) 1	Csup ^memsp(SnD, Caaid, Cbrid) 1	
					^Csup ^memsb(SnD, Caaid, Cbrid) 1	^Csup ^memsb(SnD, Caaid, Cbrid) 1	
Protocol error or Internal error		[SETDIAG] [AABrqPa] 1	[SETDIAG] [AABrqPa] 1		Csup memsp(SnD, Caaid, Cbrid) [SETDIAG] [AABrqPa] [CAFFAILi] 1	Csup memsp(SnD, Caaid, Cbrid) [SETDIAG] [AABrqPa] [CAFFAILi] 1	[SETDIAG] [AABrqPa] 1
					^Csup memsb(SnD, Caaid, Cbrid) [SETDIAG] [AABrqPa] [CAFFAILi] 1	^Csup memsb(SnD, Caaid, Cbrid) [SETDIAG] [AABrqPa] [CAFFAILi] 1	
					Csup ^memsp(SnD, Caaid, Cbrid) [SETDIAG] [AABrqPa] 1	Csup ^memsp(SnD, Caaid, Cbrid) [SETDIAG] [AABrqPa] 1	
					^Csup ^memsb(SnD, Caaid, Cbrid) [SETDIAG] [AABrqPa] 1	^Csup ^memsb(SnD, Caaid, Cbrid) [SETDIAG] [AABrqPa] 1	
C-RECOVER (ready) ind *(Continued on sheet 4 of 10)*			^Atokx, Ldretry [SETTOKX] [CRErsRTC] 3		^Atokx, Ldretry [SETTOKX] [CRErsRTC] 5		

Table A.17/X.862 – Channel (sheet 4 of 10)

State	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	Channel does not exist	Free channel, available	Free channel, not available	Channel owned by TPPM	Token awaited CAF-PLEASE req outstanding	AF-BEGIN-DIALOGUE cnf awaited CAF-PLEASE req outstanding	Clean-up
Predicates **Event**					Atwr		
C-RECOVER (ready) ind *(Concluded 2 of 2)*			^Atokx, ^Ldretry ^memsb (SnD, AAI, BI) ^memsp (SnD, AAI, BI) [SETTOKX] [CRErsUC] 3		^Atokx, ^Ldretry ^memsb (SnD, AAI, BI) [SETTOKX] [CRErsUC] 5		
			^Atokx, ^Ldretry memsb (SnD, AAI, BI) [SETTOKX] [VAtppmT] [CAFREiR] 4		^Atokx, ^Ldretry memsb (SnD, AAI, BI) Csup memsp(SnD, Caaid, Cbrid) [CAFFAILi] [VAtppmT] [SETTOKX] [CAFREiR] 4		
			^Atokx, ^Ldretry memsp (SnD, AAI, BI) [SETTOKX] [VAtppmT] [CAFREiR] 4		^Atokx, ^Ldretry memsb (SnD, AAI, BI) Csup ^memsp(SnD, Caaid, Cbrid) [VAtppmT] [SETTOKX] [CAFREiR] 4		
					^Atokx, ^Ldretry memsb (SnD, AAI, BI) ^Csup memsb(SnD, Caaid, Cbrid) [CAFFAILi] [VAtppmT] [SETTOKX] [CAFREiR] 4		
					^Atokx, ^Ldretry memsb (SnD, AAI, BI) ^Csup ^memsb(SnD, Caaid, Cbrid) [VAtppmT] [SETTOKX] [CAFREiR] 4		

Table A.17/X.862 – Channel (sheet 5 of 10)

State	1	2	3	4	5	6	7
	Channel does not exist	Free channel, available	Free channel, not available	Channel owned by TPPM	Token awaited CAF-PLEASE req outstanding	AF-BEGIN-DIALOGUE cnf awaited CAF-PLEASE req outstanding	Clean-up
Predicates Event					Atwr		
AF-RECOVER (ready) ind			^Atokx Ldretry [SETTOKX] [CRErsRTC] 3		^Atokx, Ldretry [SETTOKX] [CRErsRTC] 5		
			^Ldretry, ^Atokx ^memsb (SnD, AAI, BI) ^memsp (SnD, AAI, BI) [SETTOKX] [CRErsUC] 3		^Atokx, ^Ldretry ^memsb (SnD, AAI, BI) [SETTOKX] [CRErsUC] 5		
			^Ldretry, ^Atokx memsb (SnD, AAI, BI) [VAtppmT] [SETTOKX] [CAFREiR] 4		^Atokx, ^Ldretry memsb (SnD, AAI, BI) Csup memsp(SnD, Caaid, Cbrid) [CAFFAILi] [VAtppmT] [SETTOKX] [CAFREiR] 4		
			^Ldretry, ^Atokx memsp (SnD, AAI, BI) [VAtppmT] [SETTOKX] [CAFREiR] 4		^Atokx, ^Ldretry memsb (SnD, AAI, BI) Csup ^memsp(SnD, Caaid, Cbrid) [VAtppmT] [SETTOKX] [CAFREiR] 4		
					^Atokx, ^Ldretry memsb (SnD, AAI, BI) ^Csup memsb(SnD, Caaid, Cbrid) [CAFFAILi] [VAtppmT] [SETTOKX] [CAFREiR] 4		

(Continued on sheet 5 of 10)

Table A.17/X.862 – Channel (sheet 6 of 10)

State	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	Channel does not exist	Free channel, available	Free channel, not available	Channel owned by TPPM	Token awaited CAF-PLEASE req outstanding	AF-BEGIN-DIALOGUE cnf awaited CAF-PLEASE req outstanding	Clean-up
Predicates **Event**					Atwr		
AF-RECOVER (ready) ind *(Concluded 2 of 2)*					^Atokx, ^Ldretry memsb (SnD, AAI, BI) ^Csup ^memsb(SnD, Caaid, Cbrid) [VAtppmT] [SETTOKX] [CAFREiR] 4		
C-RECOVER (commit) ind			^Atokx, Ldretry [SETTOKX] [CRErsRTC] 3		^Atokx, Ldretry [SETTOKX] [CRErsRTC] 5		Atwr [CRErsRTC] 3
			^Atokx, ^Ldretry ^memsp (SldD, AAI, BI) ^memsp (SnD, AAI, BI) ^memsb (SnD, AAI, BI) [SETTOKX] [CRErsDC] 3		^Atokx, ^Ldretry ^memsp (SldD, AAI, BI) ^memsp (SnD, AAI, BI) [SETTOKX] [CRErsDC] 5		^Atwr [CRErsRTC] 2
			^Atokx, ^Ldretry memsp (SldD, AAI, BI) ^memsp (SnD, AAI, BI) [SETTOKX] [ARrqHrdC] 3		^Atokx, ^Ldretry memsp (SldD, AAI, BI) ^memsp (SnD, AAI, BI) [SETTOKX] [ARrqHrdC] 5		
(Continued on sheet 7 of 10)			^Atokx, ^Ldretry memsb (SnD, AAI, BI) [VAtppmT] [SETTOKX] [CAFREiC] 4		^Atokx, ^Ldretry memsp (SnD, AAI, BI) Csup memsp (SnD, Caaid, Cbrid) [CAFFAILi] [VAtppmT] [SETTOKX] [CAFREiC] 4		

Table A.17/X.862 – Channel (sheet 7 of 10)

State	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	Channel does not exist	Free channel, available	Free channel, not available	Channel owned by TPPM	Token awaited CAF-PLEASE req outstanding	AF-BEGIN-DIALOGUE cnf awaited CAF-PLEASE req outstanding	Clean-up
Predicates **Event**					Atwr		
C-RECOVER (commit) ind *(Concluded 2 of 2)*			^Atokx, ^Ldretry memsp (SnD, AAI, BI) [VAtppmT] [SETTOKX] [CAFREiC] 4		^Atokx, ^Ldretry memsp (SnD, AAI, BI) Csup ^memsp (SnD, Caaid, Cbrid) [VAtppmT] [SETTOKX] [CAFREiC] 4		
					^Atokx, ^Ldretry memsp (SnD, AAI, BI) ^Csup memsb (SnD, Caaid, Cbrid) [CAFFAILi] [VAtppmT] [SETTOKX] [CAFREiC] 4		
			^Atokx, ^Ldretry memsp (SnD, AAI, BI) [VAtppmT] [SETTOKX] [CAFREiC] 4		^Atokx, ^Ldretry memsp (SnD, AAI, BI) ^Csup ^memsb (SnD, Caaid, Cbrid) [VAtppmT] [SETTOKX] [CAFREiC] 4		
AF-RECOVER (commit) ind			^Atokx, Ldretry [SETTOKX] [CRErsRTC] 3		^Atokx, Ldretry [SETTOKX] [CRErsRTC] 5		
(Continued on sheet 8 of 10)			^Atokx, ^Ldretry ^memsp (SldD, AAI, BI) ^memsp (SnD, AAI, BI) ^memsb (SnD, AAI, BI) [SETTOKX] [CREsDC] 3		^Atokx, ^Ldretry ^memsp (SldD, AAI, BI) ^memsp (SnD, AAI, BI) [SETTOKX] [CREsDC] 5		

Table A.17/X.862 – Channel (sheet 8 of 10)

State	1	2	3	4	5	6	7
	Channel does not exist	Free channel, available	Free channel, not available	Channel owned by TPPM	Token awaited CAF-PLEASE req outstanding	AF-BEGIN-DIALOGUE cnf awaited CAF-PLEASE req outstanding	Clean-up
Predicates Event					Atwr		
AF-RECOVER (commit) ind *(Concluded 2 of 2)*			^Atokx, ^Ldretry memsp (SldD, AAI, BI) ^memsp (SnD, AAI, BI) [SETTOKX] [ARrqHrdC] 3		^Atokx, ^Ldretry memsp (SldD, AAI, BI) ^memsp (SnD, AAI, BI) [SETTOKX] [ARrqHrdC] 5		
			^Atokx, ^Ldretry memsp (SnD, AAI, BI) [VAtppmT] [SETTOKX] [CAFREiC] 4		^Atokx, ^Ldretry memsp (SnD, AAI, BI) Csup memsp(SnD, Caaid, Cbrid) [CAFFAILi] [SETTOKX] [VAtppmT] [CAFREiC] 4		
			^Atokx, ^Ldretry memsb (SnD, AAI, BI) [VAtppmT] [SETTOKX] [CAFREiC] 4		^Atokx, ^Ldretry memsp (SnD, AAI, BI) Csup ^memsp(SnD, Caaid, Cbrid) [SETTOKX] [VAtppmT] [CAFREiC] 4		
					^Atokx, ^Ldretry memsp (SnD, AAI, BI) ^Csup memsb(SnD, Caaid, Cbrid) [CAFFAILi] [SETTOKX] [VAtppmT] [CAFREiC] 4		
					^Atokx, ^Ldretry memsp (SnD, AAI, BI) ^Csup ^memsb(SnD, Caaid, Cbrid) [SETTOKX] [VAtppmT] [CAFREiC] 4		

Table A.17/X.862 – Channel (sheet 9 of 10)

State	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	Channel does not exist	Free channel, available	Free channel, not available	Channel owned by TPPM	Token awaited CAF-PLEASE req outstanding	AF-BEGIN-DIALOGUE cnf awaited CAF-PLEASE req outstanding	Clean-up
Predicates **Event**					Atwr		
C-RECOVER (retry-later) cnf or C-RECOVER (unknown) cnf							Atwr 3
							^Atwr 2
AF-TOKEN-GIVE (two-way-recovery) ind			Atwr, Atokx [VCtokrF] [VAtokxF] 2		Csup memsp(SnD, Caaid, Cbrid) [VAtokxF] [VCtokrF] [VAtppmT] [CAFGIVi] 4		
			Atwr, ^Atokx [VCtokrF] 2		Csup ^memsp(SnD, Caaid, Cbrid) [VAtokxF] [VCtokrF] 2		
					^Csup memsb(SnD, Caaid, Cbrid) [VAtokxF] [VCtokrF] [VAtppmT] [CAFGIVi] 4		
					^Csup ^memsb(SnD, Caaid, Cbrid) [VAtokxF] [VCtokrF] 2		
AF-TOKEN-PLEASE ind		Atwr [ATOKGrqTWR] 3					Atwr, ^Ptok 7

Table A.17/X.862 – Channel (sheet 10 of 10)

State	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	Channel does not exist	Free channel, available	Free channel, not available	Channel owned by TPPM	Token awaited CAF-PLEASE req outstanding	AF-BEGIN-DIALOGUE cnf awaited CAF-PLEASE req outstanding	Clean-up
Predicates **Event**					Atwr		
CAF-PLEASE req	^Ldtwr [VAtwrF] [VAtppmF] [SETAAID] [ABDrqRO] 6	[VAtppmT] [CAFGIVi] 4	Atwr, ^Atokx ^Ctokr [VCtokrT] [ATOKPrq] 5		5	6	
	Ldtwr [VAtwrT] [VAtppmF] [SETAAID] [ABDrqRT] 6		Atwr, Atokx 5				
			Atwr, Ctokr 5				
CAF-DETACH (type = free) req				Atwr, Ptok [VAtppmF] 2			
				Atwr, ^Ptok [VAtppmF] 3			
				^Atwr, Cinit [VAtppmF] 2			
				^Atwr, ^Cinit [VAtppmF] 3			
CAF-DETACH (type = not-used) req				Atwr [VAtppmF] 2			
				^Atwr, Cinit [VAtppmF] 2			
				^Atwr, ^Cinit [VAtppmF] 3			
CAF-DETACH (type = clean-up) req				[VAtppmF] 7			
Terminate-channel		Ptok [AEDrqF] [SDETrqF] 1					

End of Table A.17/X.862

Table A.18/X.862 – SACF (sheet 1 of 17)

State	1	1.1	1.2	2	3	4	5.1	5.2	6	7	8	9
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates Event		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
SAF-DETACH-ASSOCIATION (free) req				Aq [VAdtT] 2								
				^Aq [RETTOKEN] 1	[VAdtT] 3	^Aq [RETTOKEN] 1			[RETTOKEN] 1			
SAF-DETACH-ASSOCIATION (begin-fear) req				Aq [VAfT] [VAdtT] 2								
				^Aq [VAfT] [RETTOKEN] 1					[VAfT] [RETTOKEN] 1			
SAF-DETACH-ASSOCIATION (rollback-indication-expected, retain-queue = true) req									[VAdtT] 7			
SAF-DETACH-ASSOCIATION (rollback-indication-expected, retain-queue = false) req									Aq [VAdtT] [VAqF] [DISCARDQ] 1			
									^Aq [VAdtT] 7			
SAF-DETACH-ASSOCIATION (rollback-confirm-expected, retain-queue = true) req									[VAdtT] 9			
SAF-DETACH-ASSOCIATION (rollback-confirm-expected, retain-queue = false) req									Aq [VAdtT] [VAqF] [DISCARDQ] 1			
									^Aq [VAdtT] 9			
SAF-DETACH-ASSOCIATION (begin-indication-expected) req									[VAdtT] 8			

Table A.18/X.862 – SACF (sheet 2 of 17)

State	1	1.1	1.2	2	3	4	5.1	5.2	6	7	8	9
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates Event		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
SAF-SOLICIT-DIALOGUE req	^Aw Ptok [SETLPI] [ASOLrqtg] 5.1											
	^Aw ^Ptok [SETLPI] [ASOLrqd] 5.1											
AF-SOLICIT-DIALOGUE ind	Aw LPI =Alpi [SSOLi] 5.2			Aw 2						Aw 7		
	Aw LPI ^=Alpi 1											
SAF-SOLICIT-DIALOGUE rsp								Aw [ASOLrs] 1				
AF-SOLICIT-DIALOGUE cnf							^Aw [SSOLc] 1					
AF-BEGIN-DIALOGUE req	^Aw Abm [BIDREQ] [VAdcNEW] [SETCORR] [VAqT] [VAdtF] [QUEUE] 3											
(Continued on sheet 3 of 17)	^Aw Af [BIDREQ] [VAdcNEW] [SETCORR] [VAqT] [VAdtF] [QUEUE] 3											

Table A.18/X.862 – SACF (sheet 3 of 17)

State	1	1.1	1.2	2	3	4	5.1	5.2	6	7	8	9
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates Event		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
AF-BEGIN-DIALOGUE req *(Concluded 2 of 2)*	^Aw Ldbid [BIDREQ] [VAdcNEW] [SETCORR] [VAqT] [VAdtF] [QUEUE] 3											
	^Aw ^Abm,^Af ^Ldbid [VAdcNEW] [SETCORR] [SETLPI] [PASSTHRU] [VAdtF] 2											
	Aw Af [VAdcNEW] [SETCORR] [VAlpiCORR] [VAnfdT] [VAqT] [VAdtF] [QUEUE] 2											
	Aw ^Af [VAdcNEW] [SETCORR] [VAlpiCORR] [VAnfdT] [VAdtF] [PASSTHRU] 2							[VAdcNEW] [SETCORR] [VAlpiCORR] [VAnfdT] [VAdtF] [PASSTHRU] 2				

Table A.18/X.862 – SACF (sheet 4 of 17)

State	1	1.1	1.2	2	3	4	5.1	5.2	6	7	8	9
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates **Event**		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
AF-BEGIN-DIALOGUE ind	Aw ^Abm, ^Ldres Ptok Anfd, LPI=Alpi CFU [VAdcCORR] [SETCORR] [ATOKGrqKP] [VAdtF] [ATTACHMACF] [PASSTHRU] 6		[VAdcCORR] [ATTACHMACF] [PASSTHRU] 6									
	Aw ^Abm, ^Ldres ^Ptok Anfd, LPI=Alpi CFU [VAdcCORR] [VAtokrT] [ATTACHMACF] [PASSTHRU] [VAdtF] 6											
	Aw ^Abm, ^Ldres Anfd, LPI=Alpi ^CFU [VAdcCORR] [ATTACHMACF] [PASSTHRU] [VAdtF] 6											
	Aw ^Abm, Ldres Anfd, LPI=Alpi [VAdtF] [ABDrsRPdAR] 1											
(Continued on sheet 5 of 17)	Aw ^Abm Anfd, LPI^=Alpi [VAdtF] 1			Aw ^Abm 2								

Table A.18/X.862 – SACF (sheet 5 of 17)

State	1	1.1	1.2	2	3	4	5.1	5.2	6	7	8	9
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates **Event**		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
AF-BEGIN-DIALOGUE ind *(Continued 2 of 3)*	Aw ^Abm, ^Ldres Ptok ^Anfd CFU [VAdcCORR] [SETCORR] [ATOKGrqKP] [VAdtF] [ATTACHMACF] [PASSTHRU] 6											
	Aw ^Abm, ^Ldres ^Ptok ^Anfd CFU [VAdcCORR] [VAtokrT] [ATTACHMACF] [PASSTHRU] [VAdtF] 6											
	Aw ^Abm, ^Ldres ^Anfd ^CFU [VAdcCORR] [ATTACHMACF] [PASSTHRU] [VAdtF] 6											
(Continued on sheet 6 of 17)	Aw ^Abm ^Anfd Ldres [ABDrsRPdAR] [VAdtF] 1											

Table A.18/X.862 – SACF (sheet 6 of 17)

State	1	1.1	1.2	2	3	4	5.1	5.2	6	7	8	9
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates **Event**		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
AF-BEGIN-DIALOGUE ind *(Concluded 3 of 3)*	^Aw [VAnfdT] [VAlpiCORR] [VAdcCORR] [ATTACHMACF] [PASSTHRU] [VAdtF] 6			^Aw, ^Adt [VAnfdT] [DISCARDQ] [SALi] [RESETS] [VAlpiCORR] [VAdcCORR] [ATTACHMACF] [PASSTHRU] 6	^Adt [VAnfdT] [DISCARDQ] [SALi] [RESETS] [VAlpiCORR] [VAdcCORR] [ATTACHMACF] [PASSTHRU] 6		^Aw [VAnfdT] [VAlpiCORR] [VAdcCORR] [ATTACHMACF] [PASSTHRU] [VAdtF] 6					
				^Aw, Adt [VAnfdT] [DISCARDQ] [RESETS] [VAlpiCORR] [VAdcCORR] [ATTACHMACF] [PASSTHRU] 6	Adt [VAnfdT] [DISCARDQ] [RESETS] [VAlpiCORR] [VAdcCORR] [ATTACHMACF] [PASSTHRU] 6							
AF-BEGIN-DIALOGUE (accepted) rsp or AF-BEGIN-DIALOGUE (rejected(provider)) rsp or AF-BEGIN-DIALOGUE (rejected(user), dataRI) rsp or AF-BEGIN-DIALOGUE (rejected(user), rollbackRC) rsp									[SETCORR] [PASSTHRU] 6			
AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) rsp									[VAdruT] [SETCORR] [COPY] [DISCARDS] [PASSTHRU] 6			
AF-BEGIN-DIALOGUE (diagnostic = association-reserved, dataRI) cnf	1			DC^=Adc 2	3			5.2				
				^Aw DC=Adc [PASSTHRU] 6								

Table A.18/X.862 – SACF (sheet 7 of 17)

State	**1**	**1.1**	**1.2**	**2**	**3**	**4**	**5.1**	**5.2**	**6**	**7**	**8**	**9**
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates **Event**		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
AF-BEGIN-DIALOGUE (diagnostic ^= association-reserved, dataRI) cnf	1			DC^=Adc 2	3							
				DC=Adc [PASSTHRU] 6		DC=Adc [PASSTHRU] 6						
AF-BEGIN-DIALOGUE (diagnostic ^= association-reserved, rollbackRI) cnf or AF-BEGIN-DIALOGUE (diagnostic ^= association-reserved, rollbackRC) cnf				DC=Adc [PASSTHRU] 6		DC=Adc [PASSTHRU] 6						
AF-BID (token-requested = FALSE) ind	Aw Anfd, LPI=Alpi Ldres [ABIDrsR] [VAdtF] 1			Aw 2						Aw 7		Aw 9
	Aw Anfd, LPI=Alpi ^Ldres [ABIDrsA] [VAdtF] 1.2											
	Aw Anfd, LPI^=Alpi [VAdtF] 1											
	Aw ^Anfd Ldres [ABIDrsR] [VAdtF] 1											
	Aw ^Anfd ^Ldres [ABIDrsA] [VAdtF] 1.2											

Table A.18/X.862 – SACF (sheet 8 of 17)

State	1	1.1	1.2	2	3	4	5.1	5.2	6	7	8	9
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates **Event**		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
AF-BID (token-requested = TRUE) ind	Aw Anfd, LPI=Alpi Ldres [ABIDrsR] [VAdtF] 1			Aw 2						Aw 7		Aw 9
	Aw Anfd, LPI=Alpi ^Ldres, Ptok [ABIDrsA] [ATOKGrqRG] [VAdtF] 1.2											
	Aw Anfd, LPI=Alpi ^Ldres, ^Ptok ^Lddel [ABIDrsA] [VAtokrT] [VAdtF] 1.2											
	Aw Anfd, LPI=Alpi ^Ldres, ^Ptok Lddel [VAtokrT] [VAdtF] 1.1											
	Aw Anfd, LPI^=Alpi [VAdtF] 1											
	Aw ^Anfd Ldres [ABIDrsR] [VAdtF] 1											

(Continued on sheet 9 of 17)

Table A.18/X.862 – SACF (sheet 9 of 17)

State	1	1.1	1.2	2	3	4	5.1	5.2	6	7	8	9
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates Event		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
AF-BID (token-requested = TRUE) ind *(Concluded 2 of 2)*	Aw ^Anfd ^Ldres, Ptok [ABIDrsA] [ATOKGrqRG] [VAdtF] 1.2											
	Aw ^Anfd ^Ldres, ^Ptok ^Lddel [ABIDrsA] [VAtokrT] [VAdtF] 1.2											
	Aw ^Anfd ^Ldres, ^Ptok Lddel [VAtokrT] [VAdtF] 1.1											
AF-BID (accepted) cnf					Acbegq [FLUSHPAR] [VAfF] 4							
					^Acbegq, ^Adt [FLUSHALL] [VAfF] [VAqF] 4							
					^Acbegq, Adt [FLUSHALL] [VAfF] [VAqF] [VAdtF] 1							

Table A.18/X.862 – SACF (sheet 10 of 17)

State	1	1.1	1.2	2	3	4	5.1	5.2	6	7	8	9
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates Event		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
AF-BID (rejected) cnf					Adt [DISCARDQ] [VAfF] [RESETS] 1							
					^Adt [DISCARDQ] [VAfF] [RESETS] [SALi] 1							
AF-END-DIALOGUE req				^Aq [PASSTHRU] 2	[QUEUE] 3	^Aq [PASSTHRU] 4			[PASSTHRU] 6			
				Aq [QUEUE] 2		Aq [QUEUE] 4						
AF-U-ERROR req or AF-ABORT (dataRI) req or AF-GRANT-CONTROL req or AF-REQUEST-CONTROL req or AF-HANDSHAKE req or AF-HANDSHAKE-AND-GRANT-CONTROL req or AF-DEFER req or AF-PREPARE req or U-ASE req or C-CANCEL req				^Aq [PASSTHRU] 2	[QUEUE] 3	^Aq [PASSTHRU] 4			^Aq [PASSTHRU] 6			
				Aq [QUEUE] 2		Aq [QUEUE] 4			Aq [QUEUE] 6			
AF-ABORT (diagnostic ^= begin-transaction-reject, rollbackRI) req or C-ROLLBACK req or AF-EARLY-EXIT req				^Aq [DISCARDS] [PASSTHRU] 2	[QUEUE] 3	^Aq [DISCARDS] [PASSTHRU] 4			^Aq [DISCARDS] [PASSTHRU] 6			
				Aq [QUEUE] 2		Aq [QUEUE] 4			Aq [QUEUE] 6			

Table A.18/X.862 – SACF (sheet 11 of 17)

State	1	1.1	1.2	2	3	4	5.1	5.2	6	7	8	9
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates Event		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
AF-ABORT (diagnostic = begin-transaction-reject, rollbackRI) req									^Aq [VAbtrT] [COPY] [DISCARDS] [PASSTHRU] 6			
AF-ABORT (rollbackRC) req						[PASSTHRU] 4			[PASSTHRU] 6			
AF-ABORT (abortRI) req				[DISCARDQ] [DISCARDS] [PASSTHRU] *	[DISCARDQ] [DISCARDS] [PASSTHRU] *	[DISCARDQ] [DISCARDS] [PASSTHRU] *			[DISCARDQ] [DISCARDS] [PASSTHRU] *			
AF-END-DIALOGUE ind or AF-END-DIALOGUE cnf or AF-U-ERROR cnf or AF-GRANT-CONTROL ind or AF-REQUEST-CONTROL ind or AF-HANDSHAKE ind or AF-HANDSHAKE cnf or AF-HANDSHAKE-AND-GRANT-CONTROL ind or AF-HANDSHAKE-AND-GRANT-CONTROL cnf or U-ASE ind	1			2	3		5.1		[PASSTHRU] 6			9
AF-U-ERROR ind	1			2	3		5.1		[PASSTHRU] 6	7		9

Table A.18/X.862 – SACF (sheet 12 of 17)

State	1	1.1	1.2	2	3	4	5.1	5.2	6	7	8	9
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates **Event**		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
AF-END-DIALOGUE rsp or AF-U-ERROR rsp or AF-HANDSHAKE rsp or AF-HANDSHAKE-AND-GRANT-CONTROL rsp or C-BEGIN rsp or C-READY req or C-NOCHANGE req or C-COMMIT req or C-COMMIT+C-BEGIN req or C-COMMIT rsp or AF-ABORT (user, commitRI) req or AF-ABORT (nochangeRC, commit) req or AF-ABORT (nochangeRC, not-determined) req or AF-ABORT (user, commitRC) req or AF-REPORT (commitRC) req or AF-REPORT (data) req or AF-REPORT (recoverDoneRC) req or AF-REPORT (rollbackRC) req or AF-ABORT-AND-REPORT (commitRC) req or AF-ABORT-AND-REPORT (data) req or AF-ABORT-AND-REPORT (rollbackRC) req or C-ROLLBACK rsp or AF-EARLY-EXIT rsp or P-TOKEN-GIVE (sync-minor) req									[PASSTHRU] 6			
AF-REPORT (rollbackRI) request or AF-ABORT-AND-REPORT (rollbackRI) req									[DISCARDS] [PASSTHRU] 6			
C-RECOVER rsp									[PASSTHRU] [VArvyrsF] 6			
AF-ABORT (provider, abortRI) ind	*	*	*	^Adt [DISCARDQ] [PASSTHRU] *	^Adt [DISCARDQ] [PASSTHRU] *	^Adt [DISCARDQ] [PASSTHRU] *	*	*	[DISCARDQ] [PASSTHRU] *	*	*	*
				Adt [DISCARDQ] *	Adt [DISCARDQ] *	Adt [DISCARDQ] *						
AF-ABORT (user, dataRI) ind	1			2	3		5.1		[PASSTHRU] 6	7		9

Table A.18/X.862 – SACF (sheet 13 of 17)

State	1	1.1	1.2	2	3	4	5.1	5.2	6	7	8	9
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates Event		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
A-ABORT ind or A-P-ABORT ind or A-RELEASE (result = affirmative) cnf	*	*	*	^Adt [PASSTHRU] *	^Adt [PASSTHRU] *	^Adt [PASSTHRU] *	*	*	[PASSTHRU] *	*	*	*
				Adt *	Adt *	Adt *						
A-ABORT req	[PASSTHRU] *	[PASSTHRU] *	[PASSTHRU] *	[DISCARDS] [DISCARDQ] [PASSTHRU] *	[DISCARDS] [DISCARDQ] [PASSTHRU] *	[DISCARDS] [DISCARDQ] [PASSTHRU] *	[PASSTHRU] *	[PASSTHRU] *	[DISCARDS] [DISCARDQ] [PASSTHRU] *	[DISCARDS] [DISCARDQ] [PASSTHRU] *	[DISCARDS] [PASSTHRU] *	[DISCARDS] [DISCARDQ] [PASSTHRU] *
A-RELEASE (result = affirmative) rsp	[PASSTHRU] *	[PASSTHRU] *	[PASSTHRU] *	[DISCARDQ] [PASSTHRU] *	[DISCARDQ] [PASSTHRU] *	[DISCARDQ] [PASSTHRU] *	[PASSTHRU] *		[DISCARDQ] [PASSTHRU] *	[DISCARDQ] [PASSTHRU] *	[PASSTHRU] *	[DISCARDQ] [PASSTHRU] *
Protocol error	[SETDIAG] [AABrqPa] *	[SETDIAG] [AABrqPa] *	[SETDIAG] [AABrqPa] *	2	Adt [SETDIAG] [AABrqPa] *	4	[SETDIAG] [AABrqPa] *	[SETDIAG] [AABrqPa] *	6	[SETDIAG] [AABrqPa] *	[SETDIAG] [AABrqPa] *	[SETDIAG] [AABrqPa] *
					^Adt 3							
AF-DEFER ind or C-BEGIN cnf or C-READY ind or C-NOCHANGE ind or C-COMMIT ind or C-NOCHANGE rsp or C-COMMIT+C-BEGIN ind or AF-ABORT (user, commitRI) ind or AF-ABORT (nochangeRC, commit) ind or AF-ABORT (nochangeRC, not-determined) ind or AF-ABORT (user, commitRC) ind or C-COMMIT cnf or C-NOCHANGE cnf or AF-REPORT (commitRC) ind or AF-REPORT (data) ind or AF-ABORT-AND-REPORT (commitRC) ind or AF-ABORT-AND-REPORT (data) ind or AF-REPORT (recoverDoneRC) ind or C-RECOVER (commit) ind or C-RECOVER cnf or AF-RECOVER (commit) ind or AF-PREPARE ind or C-CANCEL ind									[PASSTHRU] 6			

Table A.18/X.862 – SACF (sheet 14 of 17)

State	1	1.1	1.2	2	3	4	5.1	5.2	6	7	8	9
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates **Event**		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
C-RECOVER (ready) ind or AF-RECOVER (ready) ind									[PASSTHRU] [VArvyrsT] 6			
C-BEGIN req or AF-BEGIN-TRANSACTION req				Ptok [PASSTHRU] 2	[VAcbegqT] [QUEUE] 3	Ptok [PASSTHRU] 4			Ptok [PASSTHRU] 6			
				^Ptok [VAcbegqT] [VAqT] [QUEUE] 2		^Ptok [VAcbegqT] [VAqT] [QUEUE] 4						
C-BEGIN ind or AF-BEGIN-TRANSACTION ind	Af [VAfF] [DISCARDS] [CRBrq] 9			Af, ^Adt [SALi] [DISCARDQ] [RESETS] [VAfF] [DISCARDS] [CRBrq] 9	Af, ^Adt [SALi] [DISCARDQ] [RESETS] [VAfF] [DISCARDS] [CRBrq] 9		Af [VAfF] [DISCARDS] [CRBrq]		[PASSTHRU] 6		[DISCARDS] [CRBrq] 9	
				Af, Adt [DISCARDQ] [RESETS] [VAfF] [DISCARDS] [CRBrq] 9	Af, Adt [DISCARDQ] [RESETS] [VAfF] [DISCARDS] [CRBrq] 9		9					
C-ROLLBACK ind or AF-EARLY-EXIT ind				[PASSTHRU] 6		[PASSTHRU] 6			[PASSTHRU] 6	[CRBrs] [RETTOKEN] 1		Abtr [REPREQ] [RETTOKEN] [VAbtrF] 1
												Adru [REPREQ] [RETTOKEN] [VAdruF] 1
												^Abtr, ^Adru [CRBrs] [RETTOKEN] 1
AF-REPORT (rollbackRI) ind or AF-ABORT-AND-REPORT (rollbackRI) ind				[PASSTHRU] 6		[PASSTHRU] 6			[PASSTHRU] 6			

Table A.18/X.862 – SACF (sheet 15 of 17)

State	1	1.1	1.2	2	3	4	5.1	5.2	6	7	8	9
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates **Event**		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
AF-ABORT (rollbackRI) ind				[PASSTHRU] 6		[PASSTHRU] 6			[PASSTHRU] 6			[CRBrs] [RETTOKEN] [VAbtrF] [VAdruF] 1
C-ROLLBACK cnf or AF-EARLY-EXIT cnf or AF-REPORT (rollbackRC) ind or AF-ABORT (rollbackRC) ind or AF-ABORT-AND-REPORT (rollbackRC) ind				[PASSTHRU] 6		[PASSTHRU] 6			[PASSTHRU] 6			[RETTOKEN] [VAbtrF] [VAdruF] 1
C-RECOVER (ready) req or AF-RECOVER req									Ptok [PASSTHRU] 6			
									^Ptok [VAqT] [QUEUE] 6			
C-RECOVER (commit) req									Arvyrs [PASSTHRU] [VArvyrsF] 6			
									^Arvyrs Ptok [PASSTHRU] 6			
									^Arvyrs ^Ptok [QUEUE] 6			
AF-TOKEN-GIVE (regular) ind	Aw [VAfF] 1	[ABIDrsA] [ATOKGrqRG] [VAtokrF] 1.2	Atokr [ATOKGrqRG] [VAtokrF] 1.2	Aw, ^Adt Aq [FLUSHALL] [VAcbegqF] [VAqF] [VAfF] 2		Aq [FLUSHALL] [VAcbegqF] [VAqF] [VAfF] 4		5.2	Aw Atokr [SETCORR] [ATOKGrqKP] [VAtokrF] 6			
				Aw, Adt Aq [FLUSHALL] [VAcbegqF] [VAqF] [VAfF] 1								

(Continued on sheet 16 of 17)

Table A.18/X.862 – SACF (sheet 16 of 17)

State	**1**	**1.1**	**1.2**	**2**	**3**	**4**	**5.1**	**5.2**	**6**	**7**	**8**	**9**
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates **Event**		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
AF-TOKEN-GIVE (regular) ind *(Concluded 2 of 2)*			^Atokr [VAfF] 1.2	Aw ^Aq [VAfF] 2		^Aq 4						
	^Aw [ATOKGrqRG] 1			^Aw [ATOKGrqRG] 2	[ATOKGrqRG] 3		[ATOKGrqRG] 5.1					
AF-TOKEN-GIVE (keep) ind	^Aw [ATOKGrqRG] 1			^Aw DC^=Adc [ATOKGrqRG] 2	DC^=Adc [ATOKGrqRG] 3		[ATOKGrqRG] 5.1					
				^Aw DC=Adc [FLUSHALL] [VAfF] [VAqF] [VAcbegqF] 2					^Aw DC=Adc [FLUSHALL] [VAfF] [VAqF] [VAcbegqF] 6	^Aw, Aq DC=Adc [FLUSHALL] [VAfF] [VAqF] [VAcbegqF] 7		^Aw, Aq DC=Adc [FLUSHALL] [VAfF] [VAqF] [VAcbegqF] 9
AF-TOKEN-GIVE (two-way-recovery) req									Ptok [PASSTHRU] 6			
									^Ptok [VAqT] [QUEUE] 6			
AF-TOKEN-GIVE (two-way-recovery) ind									[PASSTHRU] 6			
P-TOKEN-GIVE (sync-minor) ind	Aw [VAfF] 1			Aw, ^Adt Aq [FLUSHALL] [VAcbegqF] [VAqF] [VAfF] 2								Aw [VAfF] 9
(Continued on sheet 17 of 17)				Aw, Adt Aq [FLUSHALL] [VAcbegqF] [VAqF] [VAfF] 1								

Table A.18/X.862 – SACF (sheet 17 of 17)

State	**1**	**1.1**	**1.2**	**2**	**3**	**4**	**5.1**	**5.2**	**6**	**7**	**8**	**9**
	FREE	AF-BID ind pending (winner)	AF-BID ind accepted (winner)	STRAY	BIDDING	BID CONFIRM RECEIVED	SOLICITING	SOLICITED	BUSY	CLEANUP ROLLBACK INDICATION EXPECTED	CLEANUP BEGIN INDICATION EXPECTED	CLEANUP ROLLBACK CONFIRM EXPECTED
Predicates **Event**		Aw	Aw		^Aw, Aq	^Aw	^Aw	Aw	^Adt			
P-TOKEN-GIVE (sync-minor) ind *(Concluded 2 of 2)*	^Aw [ATOKGrqRG] 1			Aw ^Aq [VAfF] 2 ^Aw [ATOKGrqRG] 2	[ATOKGrqRG] 3		^Aw [ATOKGrqRG] 5.1		[PASSTOKEN] 6			^Aw 9
AF-TOKEN-PLEASE req									^Aq [PASSTHRU] 6 Aq [QUEUE] 6			
AF-TOKEN-PLEASE ind	1			2	3				[PASSTHRU] 6			
Reject-bid		[ABIDrsR] 1										

End of Table A.18/X.862

Table A.19/X.862 – SACF local decision variables

Name	Meaning
Ldbid	**d**ecision to **bid**
Lddel	**d**ecision to **del**ay bid rsp
Ldres	**d**ecision to **res**erve the association

A.6.2.2 Definitions of SACF variables

The following variables are defined for the SACF state table. SACF variables are Boolean variables, unless otherwise specified.

Aaet (**AE**-**t**itle): contains the peer AE-title.

Abm (**b**id **m**andatory): when set to TRUE, Abm indicates that bidding is mandatory.

Abtr (**b**egin-**t**ransaction-**r**eject): when set to TRUE, Abtr indicates that an AF-ABORT (provider, begin-transaction-reject, rollbackRI) request has been received.

Acbegq (**C**-**BEG**IN re**q**uest received): when set to TRUE, Acbegq indicates that a C-BEGIN request has been received and queued.

Acopy (**copy** issued AF-service): Acopy contains the issued AF-service, in the cases where either an AF-ABORT (rollbackRI) request or an AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) response has been issued.

Adc (**d**ialogue **c**orrelator): Adc is a variable of type Integer. Adc is used to check whether an AF-BEGIN-DIALOGUE confirm is valid. The initiator sets Adc to a unique value and copies that value into the *correlator* parameter of the AF-BEGIN-DIALOGUE request. At the recipient, Adc takes the value of the *correlator* parameter provided by the AF-BEGIN-DIALOGUE indication. Upon sending an AF-BEGIN-DIALOGUE response, the *correlator* parameter carries the value kept by Adc.

Adru (**d**ialogue **r**ejected by **u**ser): when set to TRUE, Adru indicates that an AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) response has been received.

Adt (SAF-**D**E**T**ACH-ASSOCIATION request received): when set to TRUE, Adt indicates that an SAF-DETACH-ASSOCIATION request has been received.

Af (C-BEGIN **f**ear): when set to TRUE, Af denotes that a stray C-BEGIN indication may be received.

Alpi (**l**ast **p**artner **i**dentifier): Alpi is a variable of type Integer. For a contention-winner, Alpi indicates the valid value of the *correlator* parameter to be received during the next dialogue establishment. For a contention-loser, Alpi indicates the valid value of the *correlator* parameter to be sent during the next dialogue establishment.

Anfd (**n**ot the **f**irst **d**ialogue on association): when set to TRUE, Anfd indicates that either an AF-BEGIN-DIALOGUE request has been issued by a contention-winner or that an AF-BEGIN-DIALOGUE indication has been received by a contention-loser.

Aq (**q**ueue): when set to TRUE, Aq indicates that a queue is established.

Arrh (**r**eceived **r**ecovery-context-**h**andle): Arrh is a variable of type Octet string. Arrh contains the recovery-context-handle received on the association.

Arvyrs (**r**ecover**y** **r**e**s**ponse awaited): when set to TRUE, Arvyrs indicates that a C-RECOVER (ready) indication was received and a response is awaited.

Atokr (**tok**en **r**equested): when set to TRUE, Atokr indicates that the contention-loser SACF has issued an AF-BID request with *token-requested* parameter set to TRUE, and has received an AF-BID (accepted) confirm, but has not yet received the *token*. Atokr is also used by the contention-winner to indicate that the token must be sent to the contention-loser when it arrives (after an AF-BEGIN-DIALOGUE indication without an AF-BID indication).

Atokx (**tok**en e**x**pected): when set to TRUE, Atokx indicates that a C-RECOVER indication or an AF-RECOVER indication has been received on a two-way recovery channel and that the *token* is awaited.

Atppm (attached to a **TPPM**): when set to TRUE, Atppm indicates that the SACF is attached to a TPPM. When set to FALSE, Atppm indicates that the SACF is attached to a CPM.

Atwr (**t**wo-**w**ay-**r**ecovery): when set to TRUE, Atwr indicates that the channel is established in the two-way-recovery mode**.** When set to FALSE, this variable indicates that the channel is established in the one-way-recovery mode**.**

Aw (contention-**w**inner): when set to TRUE, Aw indicates that the AEI is the contention-winner.

Ldbid (**d**ecision to **bid**): when set to TRUE, Ldbid indicates that the SACF will bid, although bidding is not mandatory.

Lddel (**d**ecision to **del**ay bid rsp): when set to TRUE, Lddel indicates that the contention-winner that does not own the *token* will delay its response to an AF-BID (*token-requested* = TRUE) indication until receipt of a P-TOKEN-GIVE (sync-minor) indication.

Ldres (**d**ecision to **res**erve the association): when set to TRUE, Ldres indicates that the association is reserved by the contention-winner.

A.6.2.3 Initialization of SACF variables

SACF variables of type Boolean are initialized to FALSE, except for the following variables:

Aaet: initialized to the peer AE-title taken from the A-ASSOCIATE parameters.

Abm: initialized to TRUE if the *Bid-Mandatory* parameter of the TP-INITIALIZE-RI APDU is set to TRUE.

Aw: initialized to TRUE if the *Contention-winner-assignment* parameter is set to "initiator" upon sending the TP-INITIALIZE-RI APDU, or if the *Contention-winner-assignment* parameter is set to "recipient" upon receipt of the TP-INITIALIZE-RI PDU.

SACF variables of type Integer are initialized to zero.

SACF variables of type Octet string are initialized to EMPTY, except for the following variables:

Arrh: initialized to the value of the *Recovery-context-handle* parameter upon receipt of either the TP-INITIALIZE-RI, or TP-INITIALIZE-RC APDU. If the *Recovery-context-handle* parameter is not specified, Arrh is initialized to NULL.

A.6.3 SACF events

A.6.3.1 Internal events

Reject-bid

Abbreviation used for "Unsolicited BID reject", as defined in 10.6.1.

A.6.3.2 Synchronizing events

There is no synchronizing event defined for the SACF state table.

A.6.4 SACF actions

A.6.4.1 Actions on services

Table A.20 lists actions that issue service primitives. These actions are named according to the following convention:

First character

A	AF-
C	C-
S	SAF-

Next characters

AB	ABORT
AL	ASSOCIATION-LOST
BD	BEGIN-DIALOGUE
BID	BID
RB	ROLLBACK
SOL	SOLICIT-DIALOGUE
TOKG	TOKEN-GIVE

Service Primitive type

rq	request
i	indication
rs	response
c	confirm

Source parameter

P	Provider
U	User

Type parameter

A	Accepted
R	Rejected
RG	Regular
KP	Keep

Mapping parameter

d	dataRI
rbc	rollbackRC
tg	tokengiveRI

Diagnostic parameter

AR	association-reserved

The remaining characters qualify the action in a manner specific to the service being issued.

NOTE – For example, ABIDrsA issues an AF-BID response with the Type parameter set to "accepted".

Table A.20/X.862 – SACF actions on services

Action name	Parameter settings/service primitive issued
AABrqPa	AF-ABORT (provider, abortRI) req
ABDrsRPdAR	– set the *diagnostic* parameter to "association-reserved". AF-BEGIN-DIALOGUE (rejected(provider), dataRI) rsp
ABIDrsA	AF-BID (accepted) rsp
ABIDrsR	AF-BID (rejected) rsp
ATOKGrqKP	AF-TOKEN-GIVE (keep) req
ATOKGrqRG	AF-TOKEN-GIVE (regular) req
CRBrq	C-ROLLBACK req
CRBrs	C-ROLLBACK rsp
SALi	SAF-ASSOCIATION-LOST ind

A.6.4.2 Actions on variables

For actions that manipulate SACF variables, the following conventions apply:

First character: V

The name of the variable being set begins at the second character.

The final characters are either:

CORR	(set the correlator parameter);
F	(set to FALSE);
NEW	(set to a new unique value); or
T	(set to TRUE).

NOTE – An example is "VAbtrF", for "set Abtr to FALSE".

A.6.4.3 Actions with free-form names

[BIDREQ] (issue an AF-**BID req**uest)

Issues the correct type of AF-BID request:

- if Anfd, set *Last-Partner-Identifier* parameter to Alpi;
- if CFU, issue AF-BID (*token-requested* = TRUE) req;
- if ^CFU, issue AF-BID (*token-requested* = FALSE) req.

[COPY] (**copy** issued service)

- copy the issued service to Acopy.

[ATTACHMACF] (**attach** to the **MACF**)

- if an AF-BEGIN-DIALOGUE (Dialogue fu selected) ind was received:
  - create a new TPPM MACF;
  - attach to the TPPM;
- if an AF-BEGIN-DIALOGUE (Recovery fu selected) ind was received:
  - attach to the CPM.

[DISCARDQ] (**discard** the **q**ueue)

- if Aq:
  - discard the queue.

[FLUSHALL] (**flush all** of the queue)

- flush the queue.

[FLUSHPAR] (**flush par**t of the queue)

- flush the queue up to and excluding the C-BEGIN req.

[PASSTHRU] (**pass** the service primitive **thr**o**u**gh)

- pass the service primitive through.

[PASSTOKEN] (**pass** the **token** to the U-ASE, pass the)

- pass the received P-TOKEN-GIVE (sync-minor) ind to the U-ASE.

[QUEUE] (**queue** the service primitive)

- queue the service primitive received from the MACF.

[REPREQ] (**Rep**eat AF-**req**uest)

- issue the primitive that is in Acopy with the mapping parameter changed into rollbackRC.

[RESETS]

- set Acbegq, Adt, Aq and Atokr to FALSE.

[RETTOKEN] (**ret**urn **token**)

Returns the *token* to the contention winner:

- if Ptok and ^Aw:
    - issue AF-TOKEN-GIVE (regular) req.

[SETCORR] (**set Corr**elator parameter)

- set the *correlator* parameter to Adc.

[SETDIAG] (**set diag**nostic)

Sets the *diagnostic* parameter of the next AF- service primitive issued by the SACF:

- set the *diagnostic* parameter to "protocol-error".

[SETLPI] (**set** ***L****ast-**P**artner-**I**dentifier* parameter)

- if Anfd:
    - set the *Last-Partner-Identifier* parameter to Alpi.

A.6.5 Notational conventions

In Table A.18, the following notational conventions are used:

- CFU is a predicate which is TRUE if the functional units parameter of the service primitive in the respective event column contains the Commit functional unit or the Recovery functional unit.
- DC is a predicate which is the value of the *correlator* parameter of the service primitive in the respective event column.
- LPI is a predicate which is the value of the *Last-Partner-Identifier* parameter of the service primitive in the respective event column.
- * is used instead of a state number if, as a result of a state transition, the association ceases to exist.

A.7 Predicates

In addition to those variables defined in A.4.2 and A.6.2, the state tables make use of predicates which reflect some state of the local system. These predicates do not have to be initialized nor modified by the PM. Table A.21 lists such predicates.

Table A.21/X.862 – Predicates

Name	Meaning
Pnew	**new** TPSUI
Ptok	**tok**en owned

Pnew (**new** TPSUI): when TRUE, Pnew indicates that the TPSUI is issuing its first TP-BEGIN-DIALOGUE request and that a new TPPM has been created and is to be initialized.

Ptok (**tok**en owned): when set to TRUE, Ptok indicates that the *token* is owned. This predicate is maintained outside of this Recommendation.

Annex B

Requirements for writing U-ASEs and application contexts

U-ASEs that are to be used within the OSI TP environment (i.e. included in an OSI TP application-context) shall observe the following guidelines while operating within the bounds of a dialogue:

a) They shall document in the application context definition any direct use of ASEs or Presentation including the embedding of U-ASE APDUs in ASE APDUs or Presentation PDUs.

b) They shall document in the application context definition any direct use of ACSE that conflicts with the TP use of ACSE.

NOTE 1 – It is recommended that the U-ASE does not make direct use of ACSE; however, the U-ASE could conceivably use the A-ABORT request or A-RELEASE request. The affirmative confirmation of the A-RELEASE request has the same effect on the TPPM as if an A-ABORT request has been received; depending on when the A-ABORT service is invoked or the A-RELEASE confirm is received, the existing transaction (if any) will be rolled back or recovered as appropriate (see 8.5.9).

c) They shall not use those non-sharable Presentation Services which are used by the TP-ASE, CCR, or ACSE.

NOTE 2 – It is permissible for the U-ASE to use the Session minor synchronize service and resynchronization service, as long as resynchronization is not done to a point before the beginning of the transaction. The use of the resynchronization service by the U-ASE should not disrupt the TP services. The U-ASE may make use of the Session major or minor synchronize service when not part of a provider-supported transaction. In this case, a U-ASE is responsible for the movement of the tokens in order that an appropriate U-ASE can have the tokens at dialogue establishment time.

d) They shall not use the CCR service directly.

e) They shall ensure that they work properly within the restrictions of this protocol, without loss of semantics.

NOTE 3 – For example, a U-ASE may not initiate commitment prior to receiving all expected U-ASE information.

In particular, when CCR is part of the application-context, the protocols of the U-ASEs that utilize the synchronize minor token shall be such that the synchronize-minor token is owned by the issuer of the TP-BEGIN-TRANSACTION request.

f) The TP Service and Protocol make use of a dummy service, TP-DATA, to indicate where one or more U-ASE services may be included in the allowed sequence of TP services. For each such occurrence of TP-DATA, the U-ASE specification shall indicate the specific U-ASE service(s), and their sequencing rules that may be substituted. Thus the U-ASE specification shall expand (and complete if it is the only U-ASE) the TPSP rules specified in this Recommendation.

g) If the U-ASE uses the P-TOKEN-GIVE service and the Commitment functional unit is selected, it is possible for the SACF, as specified in this Protocol Specification, to be unable to determine whether an incoming P-TOKEN-GIVE indication is intended for the U-ASE or not. In these cases, it is specified that the SACF passes the token to the U-ASE.

With either or both of Implicit Prepare or Dynamic Commit and Shared Control, a collision between a U-ASE APDU and either a C-READY-RI APDU or C-NOCHANGE-RI APDU is possible. This can only occur if one or both of the TPSUIs are ignoring the semantics of their application protocol, and is thus a user protocol error in the sequencing. With the particular combinations of functional units mentioned, it is not always possible for the TPSP to detect such an error in time to guarantee a correctly-propagated rollback of the transaction, and if failures occur at certain times, it may not be detected at all. If the error is detected, the "user-protocol-error" diagnostic code is used on the TP-P-ABORT indications. Applications that wish the TPPM to police this situation could use Polarized Control, or not use Implicit Prepare (but the down-tree ready signal of Dynamic Commit is still vulnerable in the latter case). Alternatively, the application can police itself using other mechanisms, e.g. numbering the U-ASE PDUs and acknowledging the last received number in an application message sent just before the C-READY PDU (see ITU-T Rec. X.851 | ISO/IEC 9804, C.9.3). It should be noted that TP can **never** police misinterpretation of the internal semantics of the U-ASE PDUs, but only their sequencing.

Annex C

Scenarios

C.1 Introduction

This annex contains scenarios which may assist in the understanding of OSI TP. Examples have been chosen to illustrate the most common functions and combinations of sequences of primitives. These are, however, just a few of the many possible sequences; these examples do not by themselves place any constraints on the use of OSI TP.

Scenarios in C.2 to C.10 concern only the "basic" functional units – dialogue, chained and unchained commitment, shared and polarized control and recovery. Scenarios in C.11 to C.18 cover all functional units.

The representation of OSI TP Protocol flows and their relationship to OSI TP Service are shown in Figure C.1.

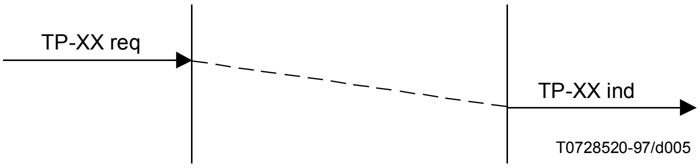


Figure C.1/X.862 – Representation of OSI TP Protocol flows

In the figures in this annex, a solid vertical line implies that there is an active dialogue and interaction can occur at any time. A dotted vertical line implies that there are still OSI TP Service interactions to complete the transaction, although the dialogue is terminated. In addition, note that some scenarios do not show complete sequences.

Table C.1 shows notations used in the scenarios for some of the CCR PDUs.

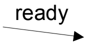
Table C.1/X.862 – Notations for certain CCR PDUs

C-NOCHANGE-RI (with-result)	A C-NOCHANGE-RI APDU with the "result-requested" parameter set to "true"
C-NOCHANGE-RI (not-required), or C-NOCHANGE-RI (n-r)	A C-NOCHANGE-RI APDU with the "result-requested" parameter set to "false"
C-NOCHANGE-RC (commit), or C-NOCHANGE-RC (cmt)	A C-NOCHANGE-RC APDU with the "atomic-action-outcome" parameter set to "committed"
C-NOCHANGE-RC (not-determined), or C-NOCHANGE-RC (not-det.)	A C-NOCHANGE-RC APDU with the "atomic-action-outcome" parameter set to "not-determined"
C-ROLLBACK-RC (EE)	A C-ROLLBACK-RC APDU with an embedded TP-EARLY-EXIT-RI APDU as User Data

The following notation is used in some scenarios to indicate the timing of logging actions: each action is formally defined in 7.3 Definitions.

R	Write a log-ready record
C	Write a log-commit record
F	Forget the transaction

T0728530-97/d006



This symbol in the headings of the scenarios involving the Dynamic Commitment functional unit indicates in which direction ready can flow on the dialogue (as set by parameters on the TP-BEGIN-DIALOGUE request and indication).

The following list gives the figure number and the title of scenarios supplied in this annex.

C.1.1 Scenarios with a single dialogue (successful cases)

C.2 Application-supported transactions with handshakes in Polarized Control

C.3 Application-supported transactions with handshakes in Shared Control

C.4 Provider-supported chained transactions

C.5 Provider-supported unchained transactions

C.6 TP-PREPARE with Data-Permitted = "false" in Polarized Control

C.7 TP-PREPARE with Data-Permitted = "true" in Polarized Control

C.8 TP-PREPARE in Shared Control

C.9 Handshake service, immediate response

C.10 Handshake service, delayed response

C.11 Deferred end dialogue, normal case

C.1.2 Scenarios with a single dialogue (unsuccessful cases)

1) Negative Response to TP-HANDSHAKE

C.12 Negative response to a TP-HANDSHAKE in Polarized Control

C.13 Collision between a TP-HANDSHAKE and a TP-U-ERROR in Polarized Control

C.14 Negative response to a TP-HANDSHAKE in Shared Control

C.15 Collision between a TP-HANDSHAKE and a TP-U-ERROR in Shared Control

2) TP-ROLLBACK scenarios

C.16 TP-ROLLBACK with Chained Transactions

C.17 TP-ROLLBACK with Unchained Transactions

3) Rejection of a TP-BEGIN-TRANSACTION

C.18 Rejection of a TP-BEGIN-TRANSACTION

4) TP-U-ERROR collision with TP-DATA

C.19 TP-U-ERROR collision with TP-DATA in Polarized Control

C.20 TP-U-ERROR collision with TP-DATA in Shared Control

5) TP-ROLLBACK with TP-U-ABORT

C.21 TP-U-ABORT response to rollback

6) TP-DEFERRED-END-DIALOGUE with TP-ROLLBACK

C.22 Deferred end dialogue cancelled by rollback in active phase

C.23 Deferred end dialogue cancelled by rollback in termination phase

7) Dialogue establishment scenarios

C.24 Rejection of the dialogue establishment, coordination level "none"

C.25 Rejection of the dialogue establishment, coordination level "commitment"

C.26 Rejection of the dialogue establishment, coordination level "commitment" (TP-COMMIT request issued)

C.27 Aborting the dialogue establishment by the requestor

C.28 Dialogue establishment followed by rollback from the requestor, dialogue rejected by recipient

C.29 Dialogue establishment followed by rollback from the requestor, dialogue accepted by recipient

C.30 Failure after a dialogue establishment

C.31 Rejection of a dialogue establishment request causing rollback

C.32 Rejection of a dialogue establishment request after TP-PREPARE

C.33 Aborting the dialogue establishment by requestor before acceptance

C.34 Aborting the dialogue establishment by requestor after acceptance

C.35 Aborting the dialogue establishment by recipient after acceptance

C.36 Rollback on a rejected dialogue

C.37 Rollback on an accepted dialogue

C.38 Failure during dialogue establishment before acceptance

C.39 Failure during dialogue establishment after acceptance

C.40 Dialogue establishment followed by rollback and subsequent communication failure

C.41 Confirmed dialogue establishment followed by rollback and subsequent user abort, recipient accepts the dialogue

C.42 Confirmed dialogue establishment followed by rollback and subsequent user abort, recipient rejects the dialogue

C.1.3 Scenarios with a single dialogue (failure case)

C.43 TP-P-ABORT during active phase

C.44 TP-P-ABORT during the first phase of commitment

C.45 TP-P-ABORT during the second phase of commitment after decision has reached subordinate

C.46 TP-P-ABORT during the second phase of commitment after the end of the transaction for the subordinate

C.47 TP-P-ABORT during the second phase of commitment before decision has reached subordinate

C.1.4 Collision scenarios on a single dialogue

C.48 Collision of TP-U-ERROR and TP-COMMIT

C.49 Collision of TP-U-ERRORs in Polarized Control

C.50 Collision of TP-U-ERRORs issued without having control

C.51 Suppression effect of TP-U-ERROR in Shared Control

C.52 Collision of TP-U-ERRORs in Shared Control

C.53 TP-U-ERROR during an outstanding handshake request

C.54 TP-U-ERROR during an outstanding handshake indication

C.55 Collision of TP-U-ERROR with TP-END-DIALOGUE

C.56 Collision of TP-U-ERRORs with TP-HANDSHAKE indication in Shared Control

C.57 Queuing TP-U-ERROR-RC before TP-BEGIN-DIALOGUE-RC is sent in Shared Control

C.58 Two TP-U-ERROR requests after a TP-HANDSHAKE collision in Shared Control

C.59 Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-END-DIALOGUE (Confirmation = "false")

C.60 Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-END-DIALOGUE (Confirmation = "true")

C.61 Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-U-ERROR in Polarized Control

C.62 Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-U-ERROR in Shared Control

C.63 Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-REQUEST-CONTROL

C.64 Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-BEGIN-TRANSACTION

C.65 Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-HANDSHAKE

C.66 Collision of a TP-COMMIT request and a TP-BEGIN-DIALOGUE confirm (rejected(provider))

C.67 Collision of a TP-COMMIT request and a TP-BEGIN-DIALOGUE confirm (rejected(user))

C.68 Collision of TP-COMMIT and TP-DATA

C.1.5 Tree with multiple dialogues (successful cases)

C.69 Commitment in a multi-dialogue tree (Chained Transactions)

C.70 Commitment in a multi-dialogue tree with usage of TP-PREPARE (Unchained Transactions)

C.1.6 Tree with multiple dialogues (unsuccessful cases)

C.71 Rollback from root node in Chained Transactions

C.72 Rollback from an intermediate node in Unchained Transactions

C.73 Rollback from an intermediate node in Chained Transactions

C.74 Two failures during first phase of commitment

C.75 Rollback-related actions (three dialogues aborted)

C.76 Failure in active phase; two isolated trees

C.77 Failure in active phase; superior tree aborted

C.78 Failure after TP-COMMIT indication issued to superior; rollback of next transaction in superior tree

C.79 Failure after TP-COMMIT indication issued to superior; isolated node refuses to be root; rollback of next transaction

C.80 Failure after all TPSUIs have received TP-COMMIT indication; rollback of next transaction in both trees

C.81 Failure after all TPSUIs have received TP-COMMIT indication; isolated node refuses to be root; rollback of next transaction in both trees

C.82 Failure after TP-COMMIT indication issued to superior; superior aborts all its dialogues; rollback of next transaction in superior tree

C.83 Failure after TP-COMMIT-COMPLETE indication issued to subordinate; rollback of next transaction in both trees

C.1.7 Heuristic decisions and reporting

C.84 Dialogue aborted during commitment; subordinate takes (wrong) decision to rollback

C.85 Dialogue aborted during commitment; subordinate takes (correct) decision to commit

C.86 Dialogue aborted during commitment; subordinate takes (wrong) decision to rollback but is able to compensate

C.87 Dialogue aborted during commitment; subordinate takes contradicting heuristic decisions producing an internal heuristic mix

C.88 Dialogue aborted during commitment; subordinate takes (wrong) decision to commit

C.89 Dialogue aborted during commitment; subordinate takes (correct) decision to rollback but a heuristic-damage is reported

C.90 Dialogue aborted before subordinate is aware of transaction termination; no heuristic decision taken but a heuristic-damage is reported

C.91 Heuristic decision and reporting in a multi-dialogue tree; all nodes above the failure receive the heuristic report

C.92 Heuristic decision and reporting in a multi-dialogue tree; root node is able to compensate and does not report heuristic mix to its TPSUI

C.93 Heuristic decision and reporting in a multi-dialogue tree; heuristic report during rollback procedure

C.1.8 Scenarios for SACF

C.94 BID used in the commit

C.95 BID not used in the commit

C.1.9 Scenarios for CPM

C.96 Channel establishment requested by contention-winner

C.97 Channel establishment requested by contention-loser (simple case)

C.98 Channel establishment requested by contention-loser (complex case)

C.99 Two-way recovery (simple case)

C.100 Recovery collision due to requests by both a superior and a subordinate

C.1.10 Read-Only scenarios

C.101 TP-READ-ONLY request with unchained transactions

C.102 TP-READ-ONLY request with chained transactions, transaction commits

C.103 TP-READ-ONLY request with chained transactions, transaction is rolled back

C.104 Dialogue abort after TP-READ-ONLY indication, chained transactions

C.105 Dialogue abort before TP-READ-ONLY indication, chained transactions

C.106 User dialogue abort after TP-READ-ONLY indication, chained transactions

C.107 TP-READ-ONLY request at intermediate and leaf, chained transactions

C.108 TP-READ-ONLY request at intermediate, leaf issues TP-COMMIT request

C.109 TP-READ-ONLY request at intermediate, leaf issues TP-ROLLBACK request

C.110 TP-READ-ONLY request at intermediate, leaf issues TP-ROLLBACK request and diagnostics

C.111 TP-READ-ONLY request at intermediate, leaf issues TP-COMMIT request but heuristically rolls back

C.112 TP-READ-ONLY request with deferred end dialogue, transaction commits

C.113 TP-READ-ONLY request with deferred end dialogue, transaction rolls back

C.114 Read-Only Intermediate promoted to Root Node

C.115 Read-Only Intermediate refuses to become Root Node

C.1.11 Early-exit scenarios

C.116 TP-EARLY-EXIT request with unchained transactions

C.117 TP-EARLY-EXIT request with chained transactions

C.118 TP-EARLY-EXIT request in response to TP-PREPARE request, unchained transactions

C.119 TP-EARLY-EXIT request collision with TP-PREPARE request, unchained transactions

C.120 Repeated use of a dialogue with early-exit and unchained transactions

C.121 TP-EARLY-EXIT request, no collision

C.122 Collision with transaction completion

C.123 Collision of early exit and rollback (1)

C.124 Collision of early exit and rollback (2)

C.125 Early Exit with one read-only subordinate

C.126 Early Exit with one read-only subordinate

C.127 Early Exit with one read-only subordinate, TP-P-ABORT indication after requesting early exit

C.128 Early Exit with one read-only subordinate, TP-P-ABORT indication after requesting early exit

C.129 Early Exit with one read-only subordinate, TP-P-ABORT indication on a subordinate dialogue after requesting early exit

C.130 Early Exit with one read-only subordinate, TP-P-ABORT indication after requesting early exit.

C.131 Early Exit in the termination phase

C.132 Early Exit with one read-only subordinate and defer-end-dialogue

C.133 Early Exit with one read-only subordinate and defer-end-dialogue in unchained.

C.134 Early Exit above a read-only dialogue with defer-grant-control in unchained, subordinate aborts the dialogue.

C.135 Early Exit and collision with defer

C.1.12 Static one-phase commitment scenarios

C.136 TP-ONE-PHASE request with unchained transactions

C.137 TP-ONE-PHASE request with chained transactions

C.138 TP-ONE-PHASE request with unchained transactions, rollback case

C.139 TP-ONE-PHASE request with unchained transactions, dialogue aborts late

C.140 TP-ONE-PHASE request with unchained transactions, dialogue aborts early

C.141 Static one-phase above two-phase, unchained transactions

C.142 Static one-phase above two-phase, chained transactions

C.143 Static one-phase above two-phase, chained transactions, leaf rolls back

C.144 Static one-phase above two-phase, chained transactions, no reporting

C.145 Static one-phase and read-only, unchained

C.146 Static one-phase and read-only, chained

C.147 Flow of C-BEGIN on static one-phase OCC-path, C-BEGIN not receivable

C.148 Flow of C-BEGIN on static one-phase OCC-path, TP-U-ABORT pending

C.149 Flow of C-BEGIN on static one-phase OCC-path, TP-U-ABORT request in Ready-state

C.150 Flow of C-BEGIN on static one-phase OCC-path, TP-U-ABORT request in Ready-state

C.1.13 Implicit prepare scenarios

C.151 Implicit prepare with unchained transactions, polarized control

C.152 Implicit prepare with unchained transactions, shared control

C.153 Implicit prepare with intermediate and read-only leaf, chained transactions

C.154 Implicit prepare and heuristic commit

C.155 Implicit prepare and heuristic rollback

C.156 Implicit prepare, heuristic commit and dialogue abort

C.1.14 TP-ROLLBACK scenarios

C.157 Rollback from superior in active phase

C.158 Rollback from subordinate in active phase

C.159 Rollback from both sides in active phase

C.160 Rollback in phase 1 of commitment

C.161 Rollback in phase 1 of commitment, heuristic damage is reported

C.162 Rollback in phase 1 of commitment, heuristic damage is not reported

C.163 Rollback in phase 1 of commitment, a completion report is provided

C.164 Rollback in phase 1 of commitment, a cancel is sent

C.165 Rollback from superior in active phase

C.166 Rollback from subordinate in active phase

C.167 Rollback in phase 1 of commitment, a completion reports are provided

C.168 Rollback in phase 1 of commitment, cancel is used

C.1.15 Dynamic Commitment scenarios

C.170 Ready down tree with TP-COMMIT request at subordinate

C.171 Ready down tree with TP-READ-ONLY request at subordinate which becomes coordinator

C.172 Ready down tree with TP-READ-ONLY request at subordinate which refuses to become coordinator

C.173 Ready down tree with TP-ONE-PHASE request at subordinate

C.174 Ready down tree only, subordinate has control

C.175 Ready either way, subordinate becomes coordinator

C.176 TP-PREPARE with Data-Permitted set to "true", followed by TP-PREPARE

C.177 Implicit prepare with dynamic one-phase commit

C.178 Collision of ONE-PHASE with ONE-PHASE

C.179 Collision of ONE-PHASE with READ-ONLY

C.180 Alternate commit initiator

C.181 Alternate commit initiator with one-phase and read-only, no reporting

C.182 Alternate commit initiator with one-phase and read-only, with heuristic reporting

C.183 One-phase commit procedure with sending of C-PREPARE-RI followed by C-NOCHANGE-RI

C.184 One-phase above one-phase

C.185 One-phase above one-phase, unchained, no reporting

C.186 One-phase everywhere

C.2 Scenarios with a single dialogue (successful cases)

The following scenarios demonstrate the service primitives in one dialogue between two TPSUIs.

C.2.1 Application-supported transactions

Two scenarios are shown. Figure C.2 illustrates Polarized Control and Figure C.3 illustrates Shared Control. Apart from that the scenarios are equivalent. In each, a dialogue is begun and data is passed between the two TPSUIs. At the midpoint, the Handshake service is used to synchronize the two partners. After further transfer of data, the partners synchronize again and end the dialogue (by using the confirmed TP-END-DIALOGUE service).

C.2.2 Provider-supported chained transactions

The scenario in Figure C.4 shows a dialogue being established between two TPSUIs to support a chained sequence of transactions. After the second transaction is completed, the dialogue is ended. The TP-DEFERRED-END-DIALOGUE service is used to request termination of the dialogue after the successful conclusion of the second transaction.

NOTE – As drawn, the scenario is accurate for Shared Control; for Polarized Control, each TP-DATA request would need to be followed by a TP-GRANT-CONTROL request to pass control of the dialogue.

C.2.3 Provider-supported unchained transactions

The scenario in Figure C.5 shows a dialogue being established between two TPSUIs to support unchained transactions. Two transactions are performed. After the second transaction is completed, the dialogue is ended. The first transaction is begun simultaneously with the dialogue (parameter Begin-Transaction of TP-BEGIN-DIALOGUE set to "true"). After the first transaction has finished, data can be exchanged between the two TPSUIs outside the scope of a transaction although this is not shown. After the second transaction has ended, the dialogue is terminated by the superior TPSUI.

NOTE – As drawn, the scenario is accurate for Shared Control; for Polarized Control, each TP-DATA request would need to be followed by a TP-GRANT-CONTROL request to pass control of the dialogue.

C.2.4 TP-PREPARE scenarios

Three scenarios are documented for TP-PREPARE:

- TP-PREPARE with Data-Permitted = "false" in Polarized Control;
- TP-PREPARE with Data-Permitted = "true" in Polarized Control;
- TP-PREPARE in Shared Control.

C.2.4.1 TP-PREPARE with Data-Permitted = "false" in Polarized Control

The scenario in Figure C.6 describes a sequence of primitives in the case when data is exchanged between two TPSUIs and a TP-PREPARE request is issued with Data-Permitted = "false" in Polarized Control.

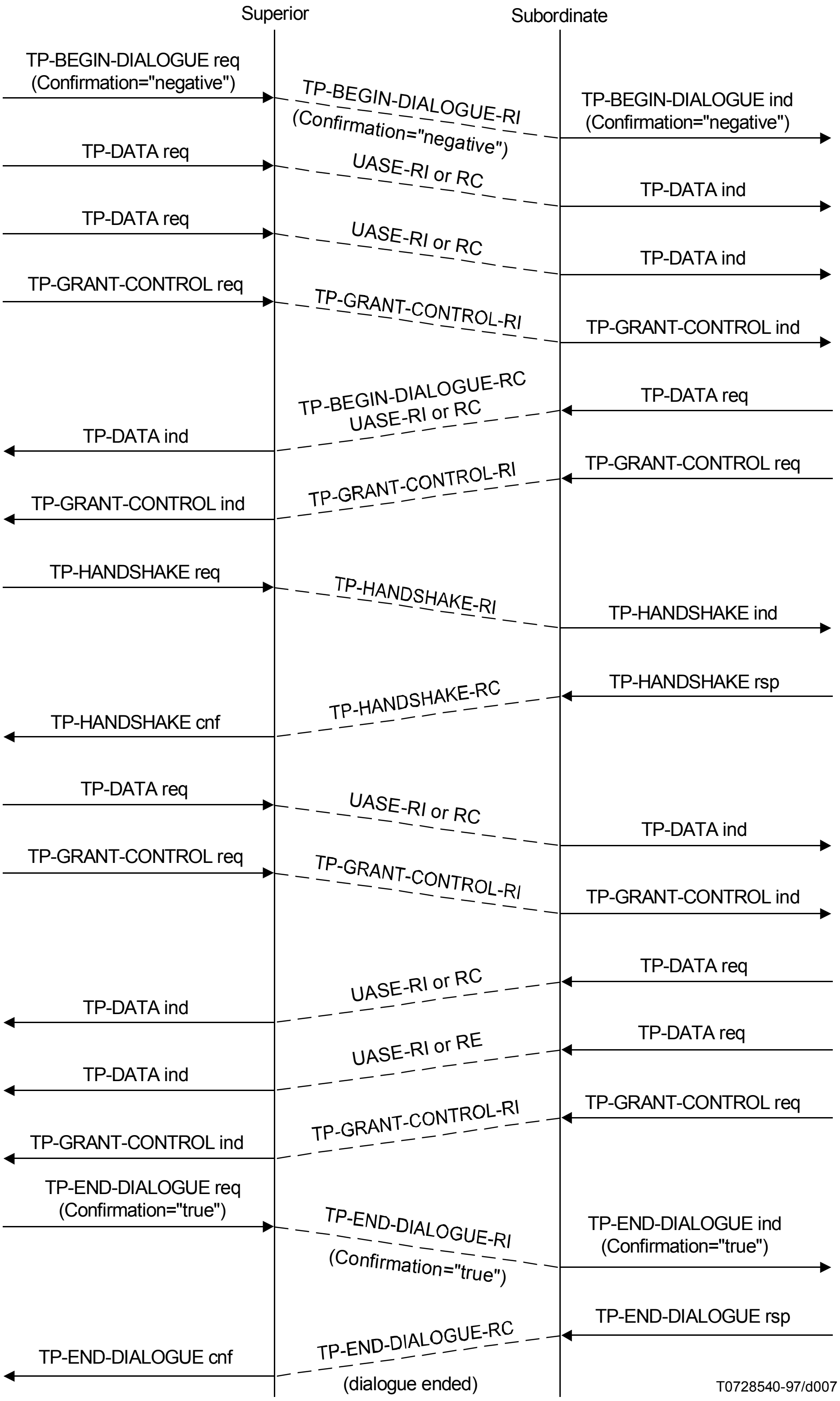


Figure C.2/X.862 – Application-supported transactions with handshakes in Polarized Control

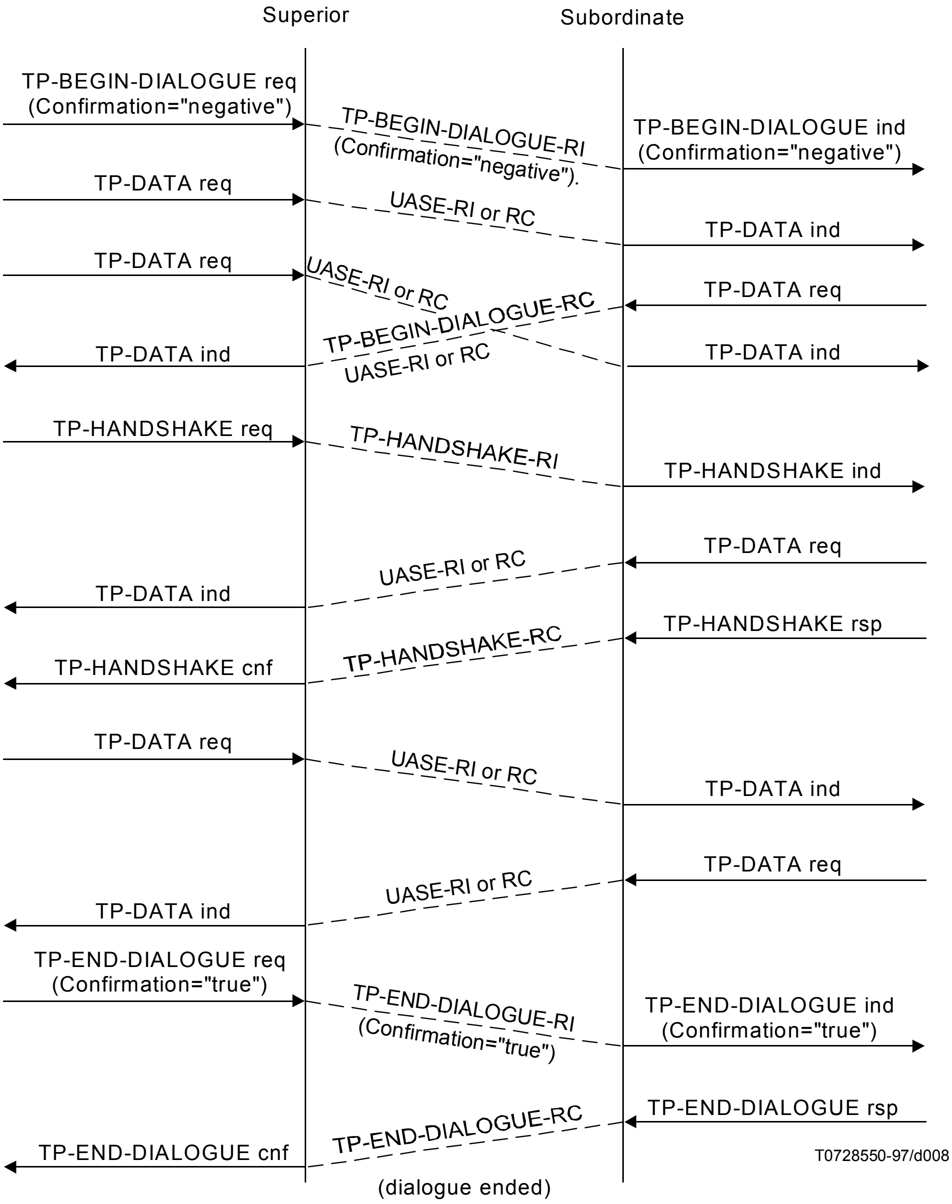


Figure C.3/X.862 – Application-supported transactions with handshakes in Shared Control

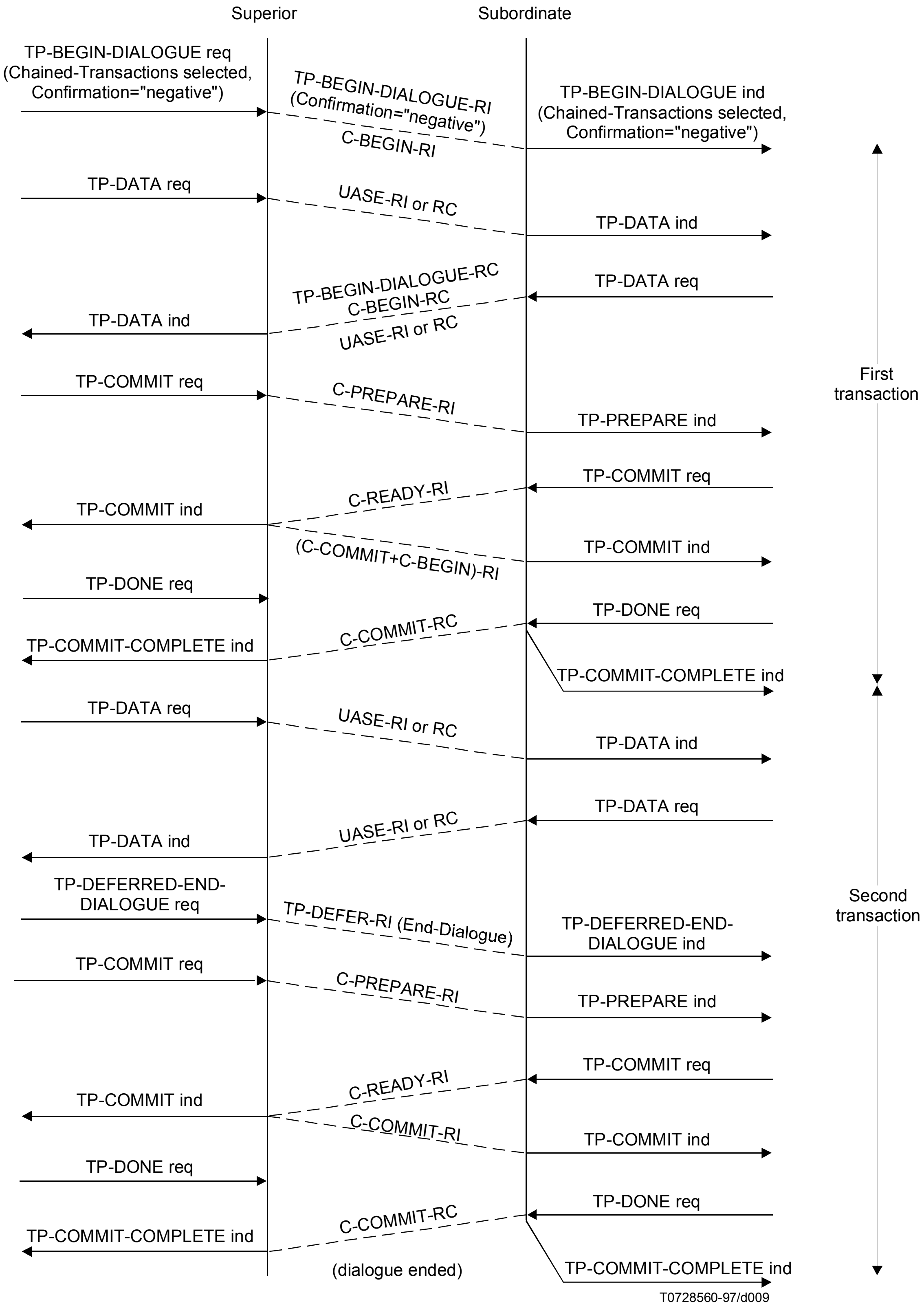


Figure C.4/X.862 – Provider-supported chained transactions

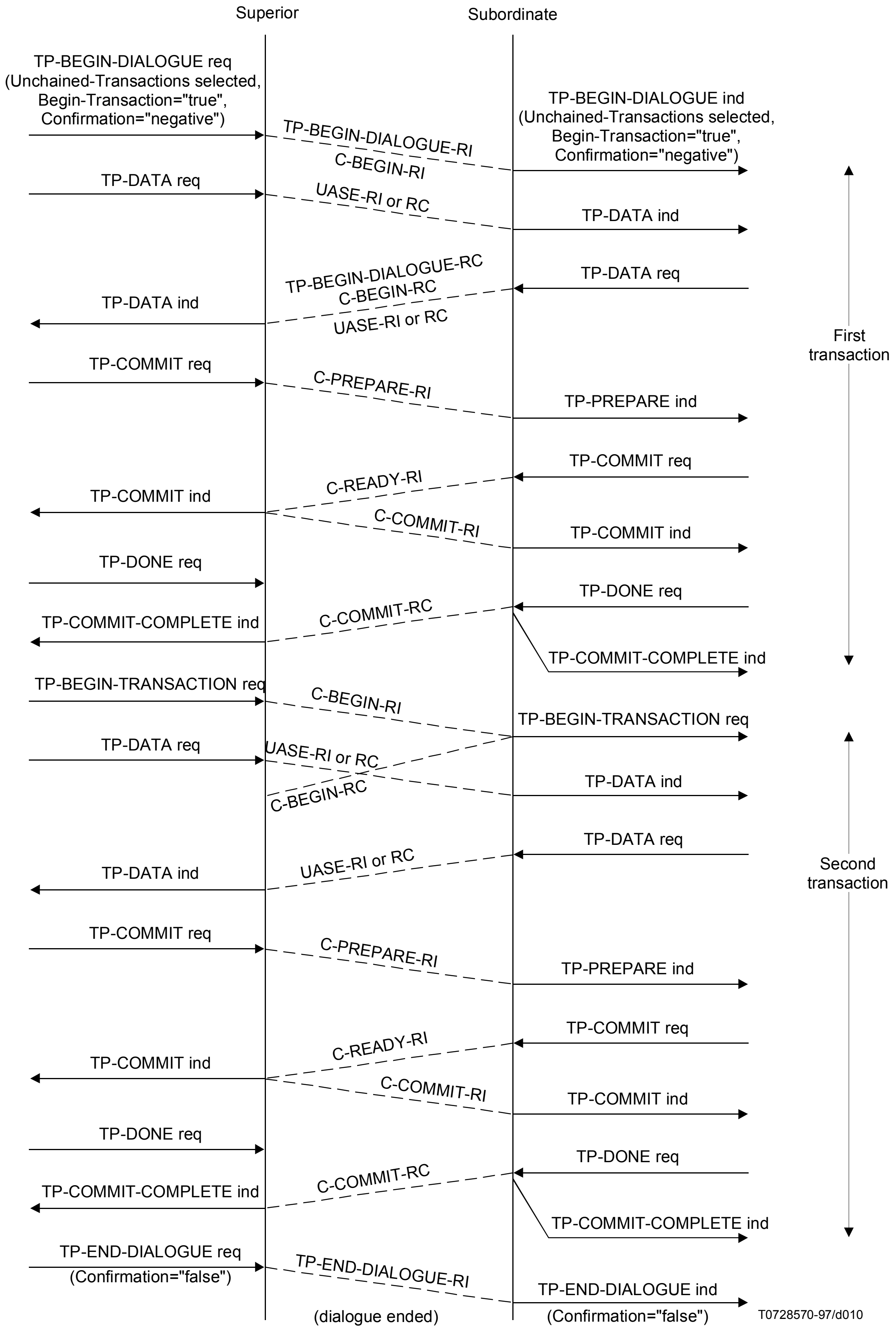


Figure C.5/X.862 – Provider-supported unchained transactions

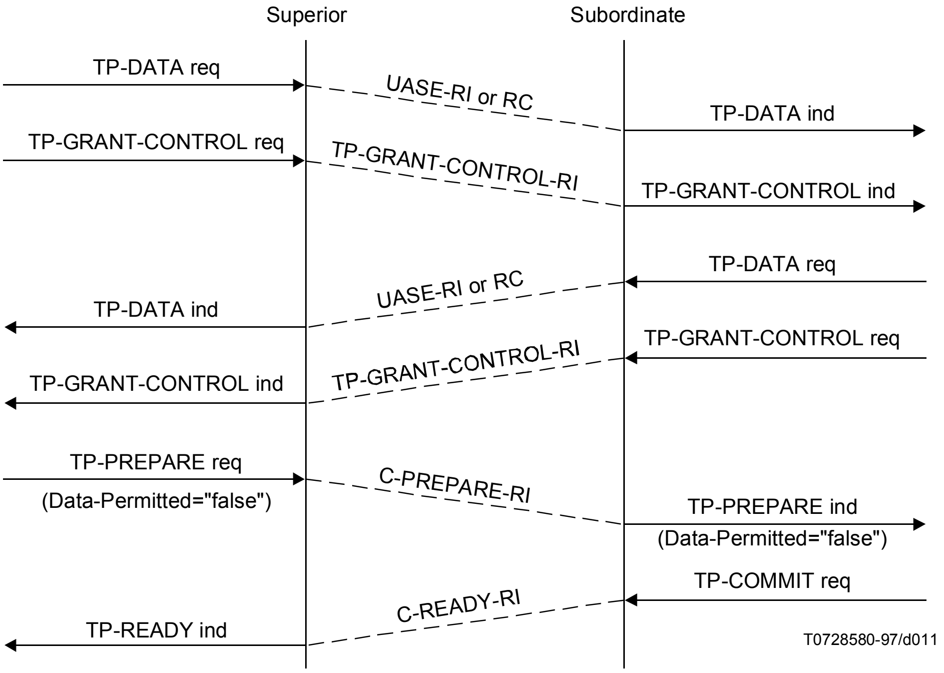


Figure C.6/X.862 – TP-PREPARE with Data-Permitted = "false" in Polarized Control

C.2.4.2 TP-PREPARE with Data-Permitted = "true" in Polarized Control

The scenario in Figure C.7 describes a similar sequence of primitives in the case when data is sent to the remote TPSUI and then a TP-PREPARE request is issued with Data-Permitted = "true" to allow the remote TPSUI to send data before issuing the TP-COMMIT request.

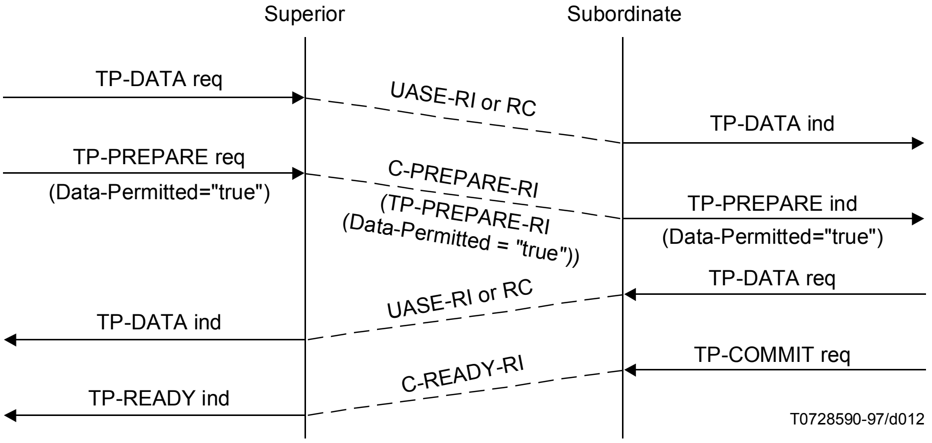


Figure C.7/X.862 – TP-PREPARE with Data-Permitted = "true" in Polarized Control

C.2.4.3 TP-PREPARE in Shared Control

The scenario in Figure C.8 describes a sequence of primitives in the case when data is exchanged between two TPSUIs and a TP-PREPARE request is issued in Shared Control.

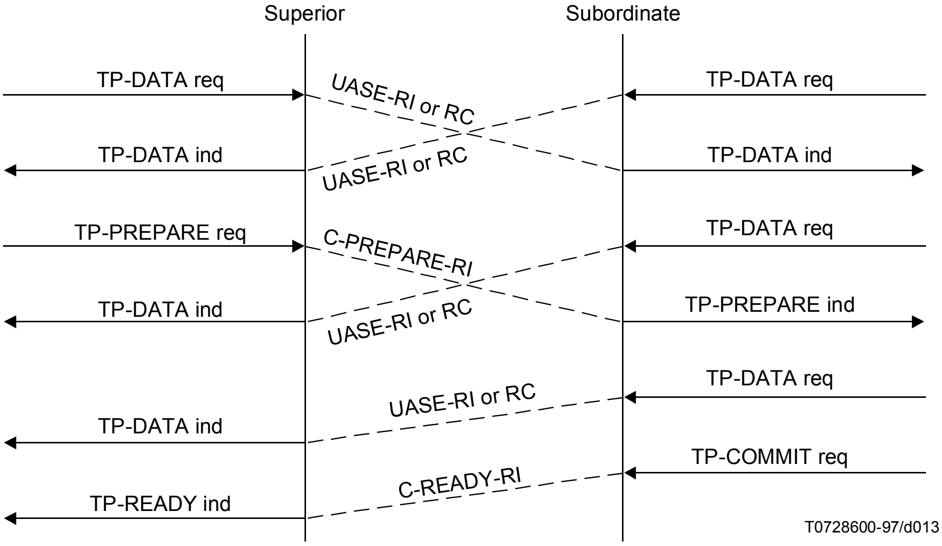


Figure C.8/X.862 – TP-PREPARE in Shared Control

C.2.5 Handshake services: illustration of Confirmation-Urgency parameter

Scenarios in Figures C.9 and C.10 demonstrate the effect of the Confirmation-Urgency parameter of the TP-HANDSHAKE request and TP-HANDSHAKE-AND-GRANT-CONTROL request services. The Confirmation-Urgency parameter is passed to the TPPM at side B but it is not made visible to the TPSUI.

– Figure C.9: When:

1) a TP-HANDSHAKE is used in Polarized Control;

2) TP-HANDSHAKE-AND-GRANT-CONTROL with Confirmation-Urgency set to "urgent"; or

3) TP-HANDSHAKE is used in Shared Control and Confirmation-Urgency set to "urgent".

The confirmation shall be issued immediately.

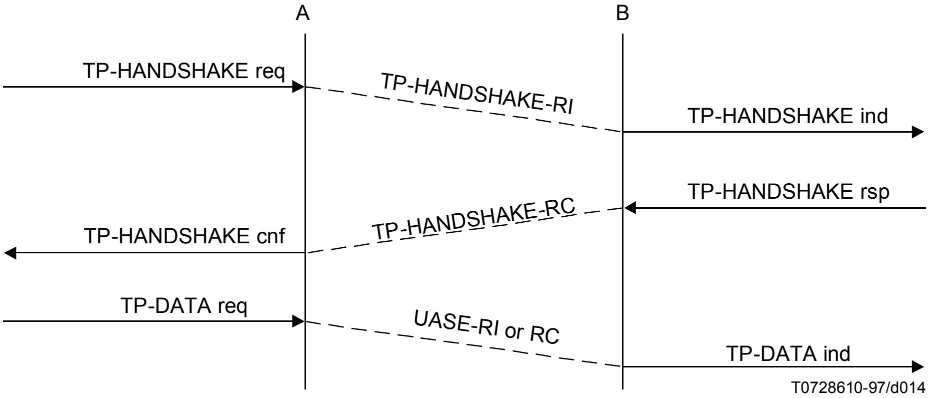


Figure C.9/X.862 – TP-HANDSHAKE – Immediate response

– Figure C.10: When TP-HANDSHAKE is used in Shared Control or when TP-HANDSHAKE-AND-GRANT-CONTROL is used, it is a local TPPM decision whether to concatenate the handshake response with other services. This could result in a substantial delay as shown by case (B). The Confirmation-Urgency parameter allows the TPSUI to specify that the handshake confirm should be issued without delay as illustrated in case (A).

NOTE – Figure C.10 applies equally to TP-HANDSHAKE-AND-GRANT-CONTROL in Polarized Control.

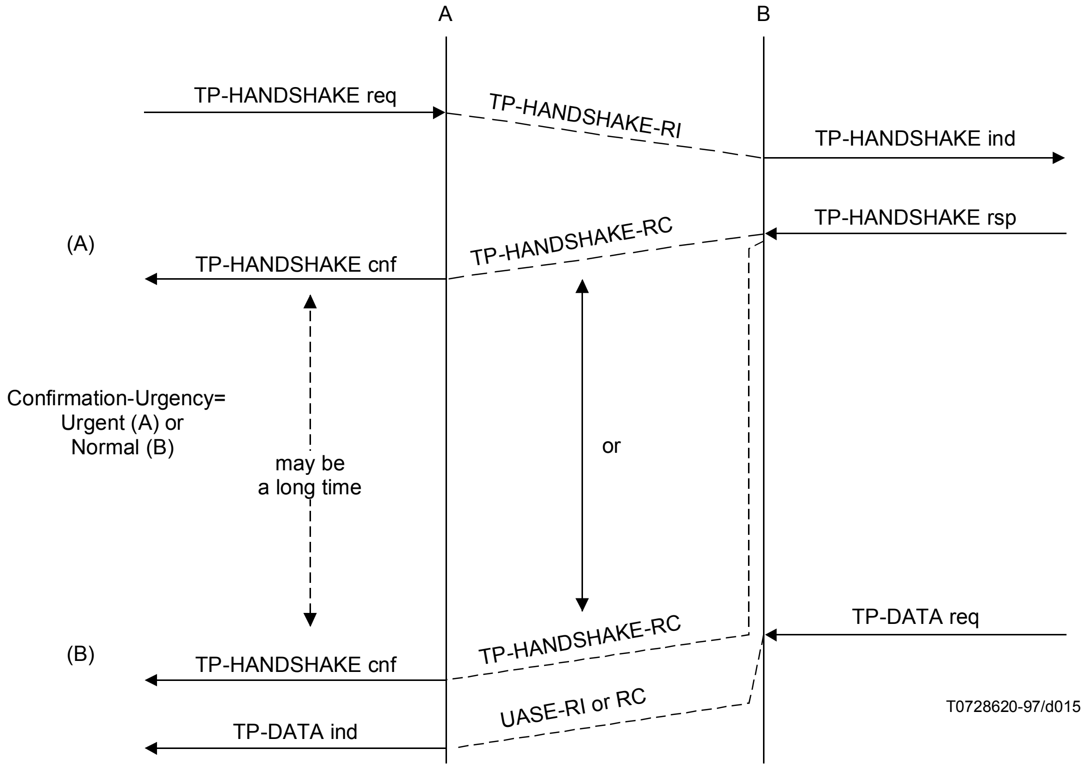


Figure C.10/X.862 – TP-HANDSHAKE – Delayed response

C.2.6 Deferred End Dialogue service

The scenario in Figure C.11 describes a sequence of primitives in the normal case of a TP-DEFERRED-END-DIALOGUE.

C.3 Scenarios with a single dialogue (unsuccessful cases)

C.3.1 Negative response to a TP-HANDSHAKE

C.3.1.1 Negative response to a TP-HANDSHAKE in Polarized Control

The scenario in Figure C.12 describes a sequence of primitives in the case when a TP-DATA request is followed by a TP-HANDSHAKE request and the TP-HANDSHAKE is negatively responded to in Polarized Control.

A collision between a TP-HANDSHAKE request and a TP-U-ERROR request is treated as an early negative response to TP-HANDSHAKE as shown in the scenario of Figure C.13.

C.3.1.2 Negative response to a TP-HANDSHAKE in Shared Control

The scenario in Figure C.14 describes a sequence of primitives in the case when a TP-DATA exchange between two TPSUIs is followed by a TP-HANDSHAKE request that is negatively responded to in Shared Control. The simple negative response to TP-HANDSHAKE in Shared Control is shown below.

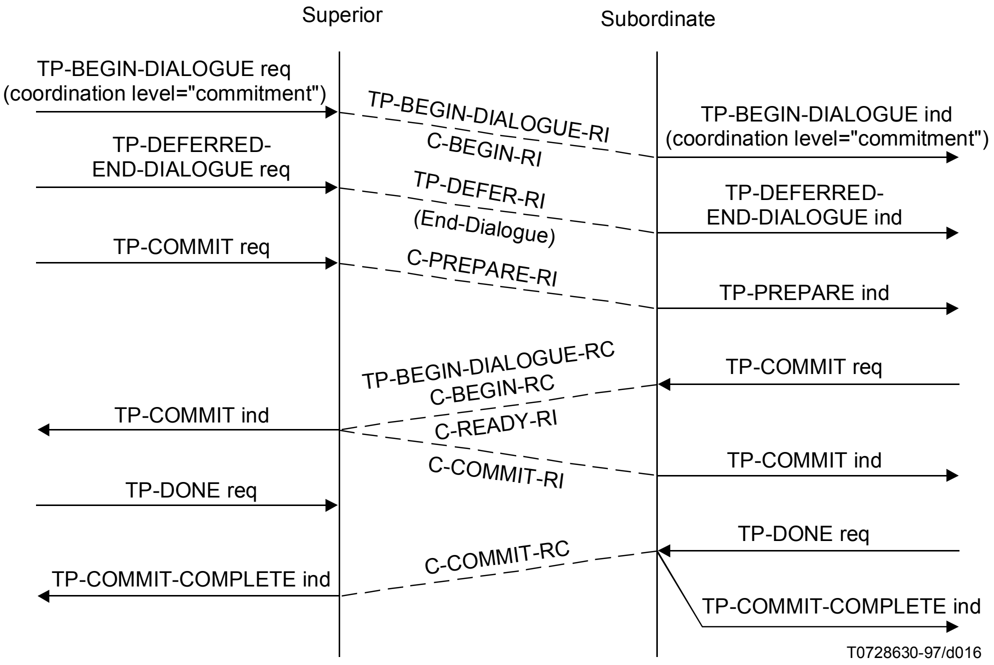


Figure C.11/X.862 – TP-DEFERRED-END-DIALOGUE – Normal case

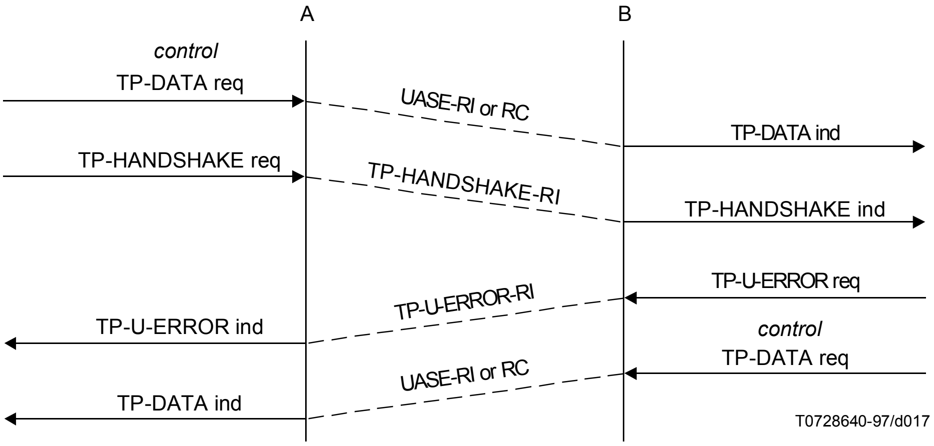


Figure C.12/X.862 – Negative response to a handshake request in Polarized Control

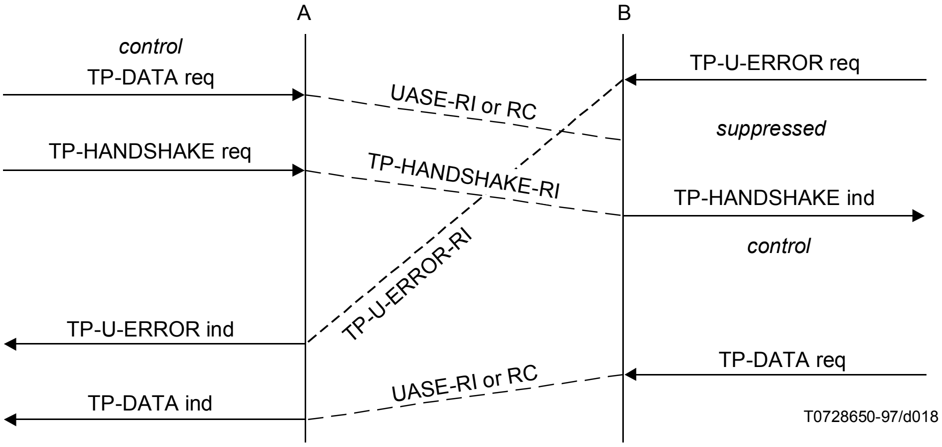


Figure C.13/X.862 – Anticipated negative response to a TP-HANDSHAKE in Polarized Control

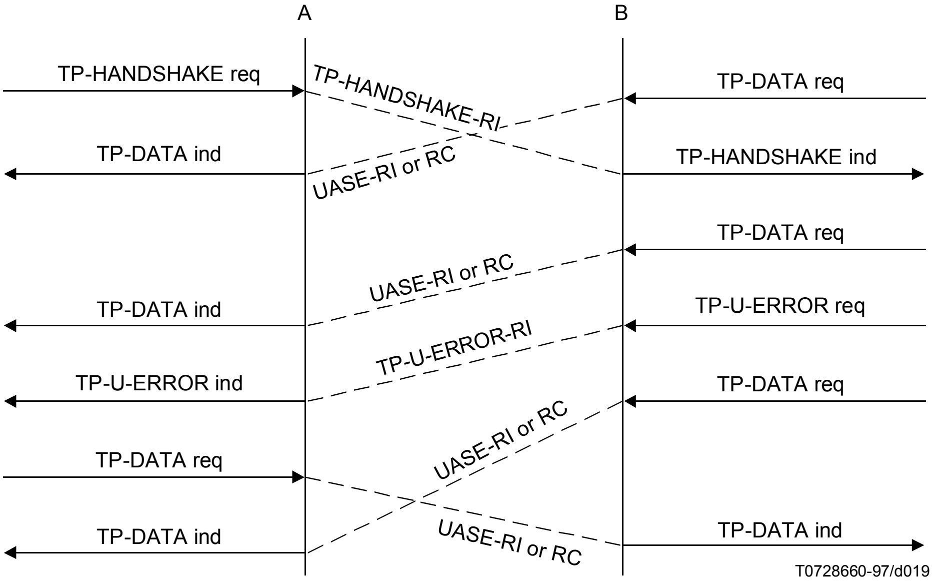
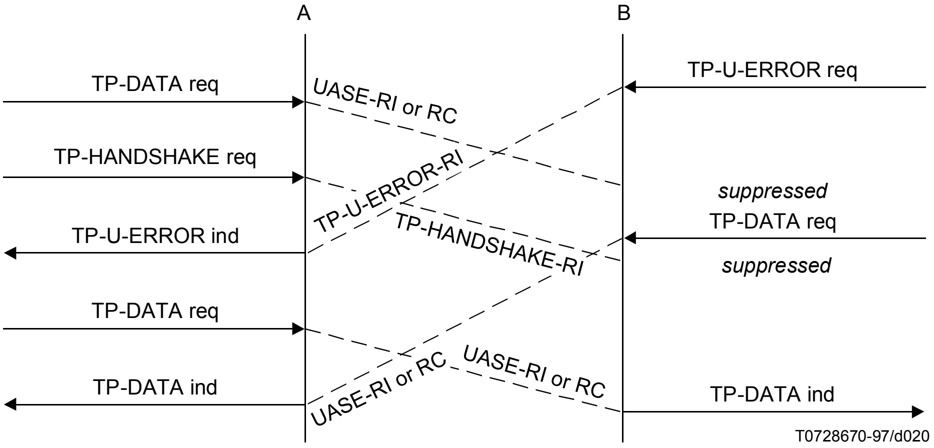


Figure C.14/X.862 – Negative response to a TP-HANDSHAKE in Shared Control

A collision between a TP-HANDSHAKE request and a TP-U-ERROR request is treated as an early negative response to TP-HANDSHAKE in Shared Control as shown in the scenario in Figure C.15. The TP-HANDSHAKE indication is suppressed because B would not know if the corresponding request had been issued before the TP-U-ERROR indication (and the TP-U-ERROR request would have been an anticipated negative response to the handshake) or after the TP-U-ERROR (and the TP-HANDSHAKE is not related to TP-U-ERROR).



NOTE – In this scenario, TP-HANDSHAKE may be replaced by TP-END-DIALOGUE with the Confirmation parameter set to "true".

Figure C.15/X.862 – Anticipated negative response to a TP-HANDSHAKE request in Shared Control

C.3.2 TP-ROLLBACK scenarios

C.3.2.1 TP-ROLLBACK with Chained Transactions

The scenario in Figure C.16 describes a sequence of primitives in the case when a transaction is rolled back by one of the TPSUIs issuing a TP-ROLLBACK request during the active phase. The TPSUIs each immediately begin a new transaction.

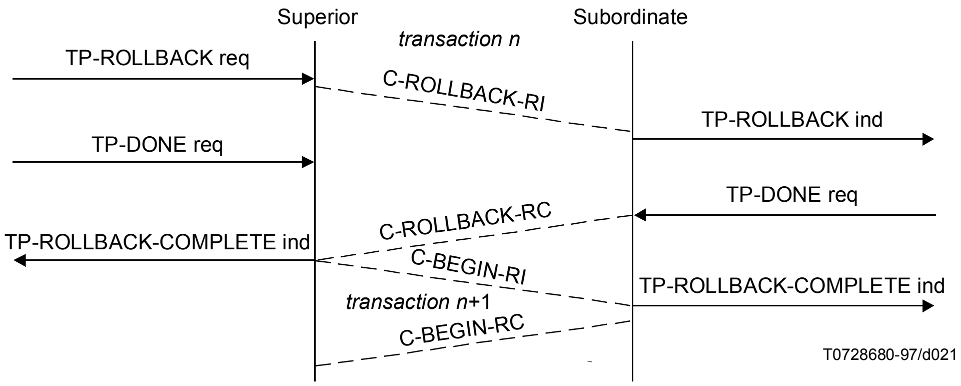


Figure C.16/X.862 – TP-ROLLBACK with the Chained Transactions functional unit

C.3.2.2 TP-ROLLBACK with Unchained Transactions

The scenario for TP-ROLLBACK with unchained transactions, shown in Figure C.17, is identical to the scenario for chained transactions except that a new transaction is not begun until a TP-BEGIN-TRANSACTION request has been issued.

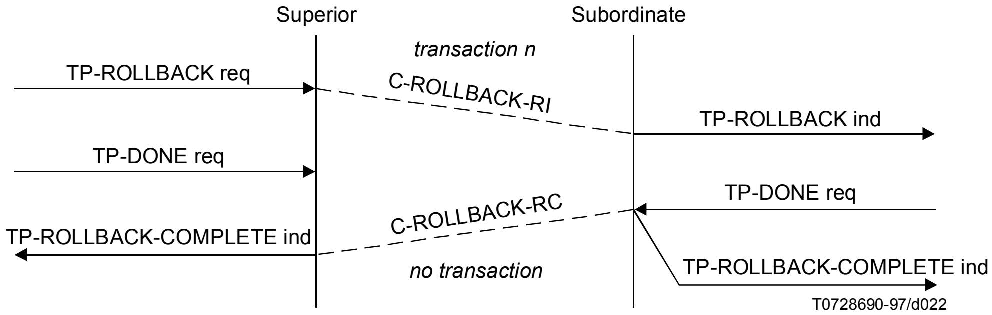


Figure C.17/X.862 – TP-ROLLBACK with the Unchained Transactions functional unit

C.3.3 Rejection of a TP-BEGIN-TRANSACTION request

The scenario in Figure C.18 describes a situation in which an attempt to include a subordinate TPSUI in the current transaction failed because the subordinate is already participating in another provider-supported transaction. TP-BEGIN-TRANSACTION is rejected, the dialogue between A and B is aborted; the transactions are not rolled back.

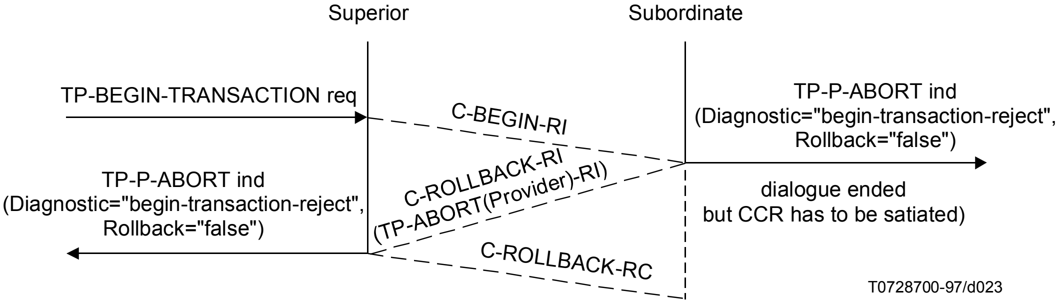


Figure C.18/X.862 – Rejection of a P-BEGIN-TRANSACTION request

C.3.4 TP-U-ERROR collision with TP-DATA

C.3.4.1 TP-U-ERROR collision with TP-DATA in Polarized Control

The scenario in Figure C.19 describes a sequence of primitives in the case of a collision between a TP-DATA request and a TP-U-ERROR request in Polarized Control. TPSUI B is not allowed to issue a TP-DATA request until the TP-GRANT-CONTROL indication is received.

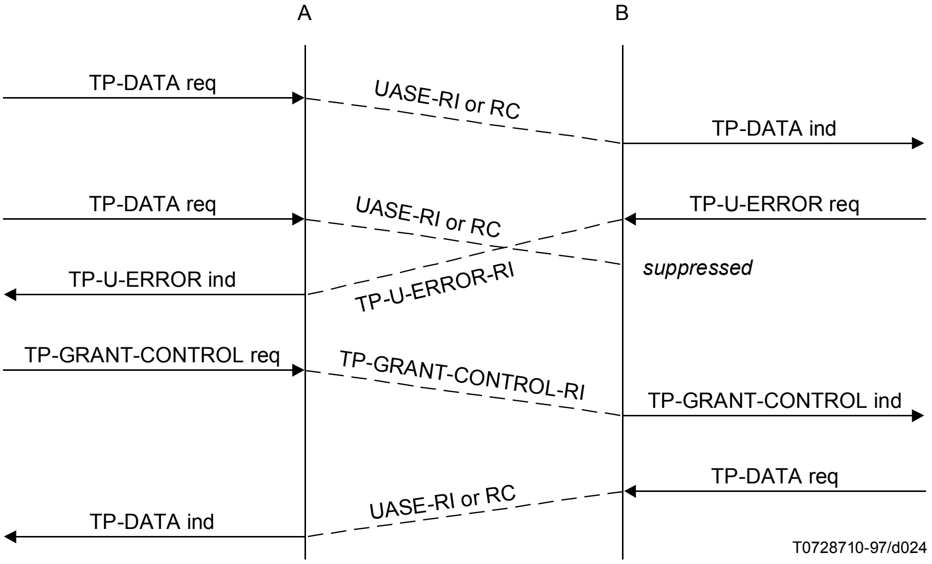


Figure C.19/X.862 –TP-U-ERROR collision with TP-DATA in Polarized Control

C.3.4.2 TP-U-ERROR Collision with TP-DATA in Shared Control

The scenario of Figure C.20 describes a sequence of primitives in the case when a TP-DATA exchange between two TPSUIs is followed by a collision between a TP-DATA request and a TP-U-ERROR request in Shared Control.

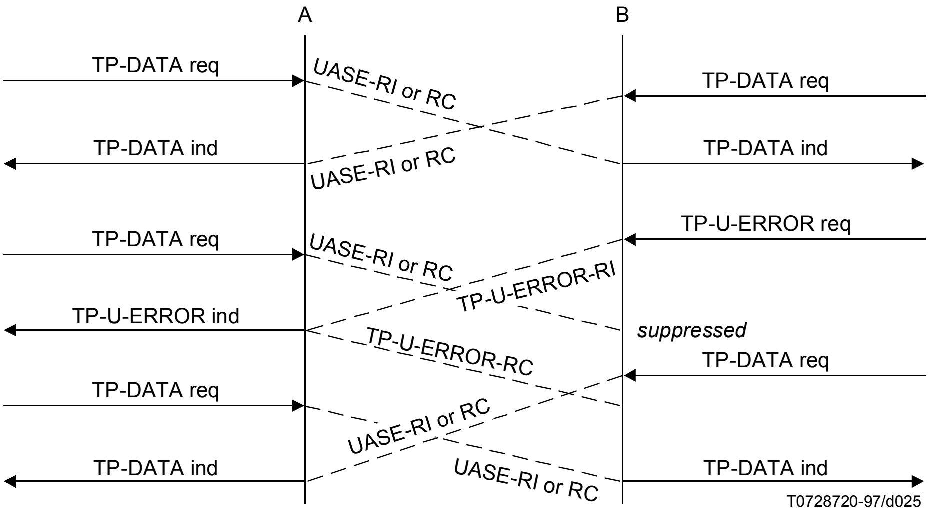


Figure C.20/X.862 – TP-U-ERROR collision with TP-DATA in Shared Control

C.3.5 TP-ROLLBACK with TP-U-ABORT

The scenario in Figure C.21 describes a sequence of primitives in the case when a TP-ROLLBACK indication is followed by a TP-U-ABORT request to abort the dialogue.

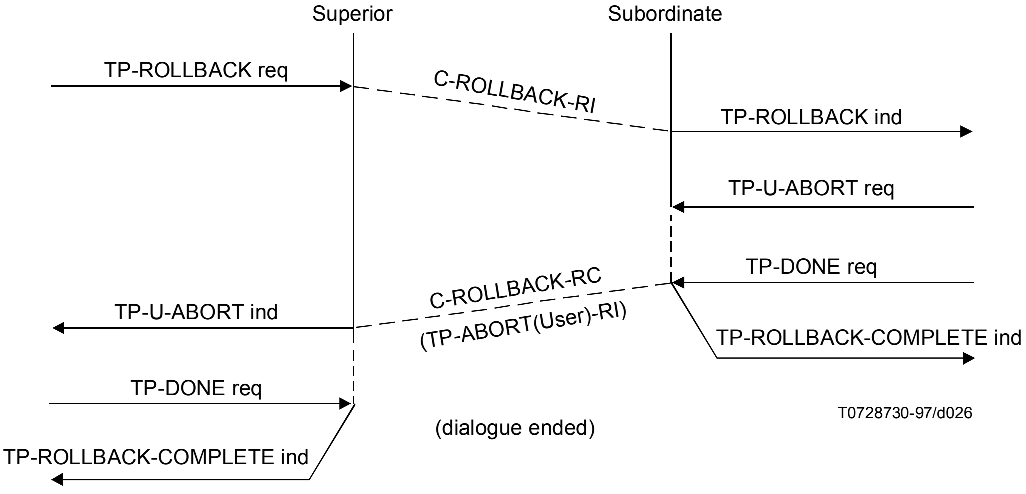
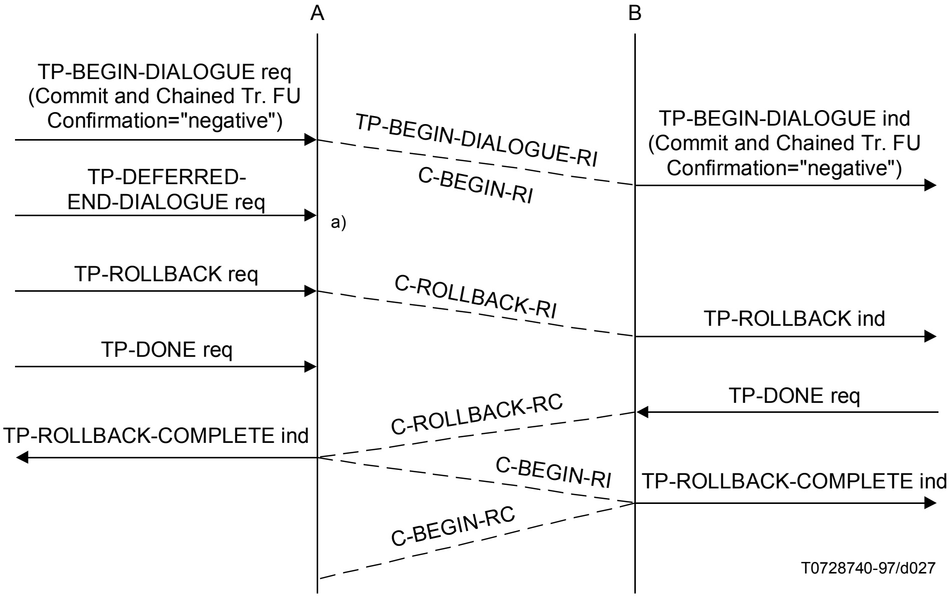


Figure C.21/X.862 – TP-U-ABORT in response to a TP-ROLLBACK indication

C.3.6 TP-DEFERRED-END-DIALOGUE with TP-ROLLBACK

C.3.6.1 TP-ROLLBACK request issued during the active phase

The scenario of Figure C.22 describes a sequence of primitives in the case when a TP-DEFERRED-END-DIALOGUE request is cancelled by a TP-ROLLBACK request during the active phase of a provider-supported transaction. The dialogue is not terminated.



[a] The TP-DEFERRED-END-DIALOGUE-RI, not sent immediately, has been discarded by the rollback procedure.

NOTE – TP-BEGIN-DIALOGUE-RC and C-BEGIN-RC are not required in this case. C-ROLLBACK-RC serves as the confirmation instead.

Figure C.22/X.862 – TP-DEFERRED-END-DIALOGUE cancelled by a rollback

C.3.6.2 TP-ROLLBACK request issued after a TP-PREPARE indication

The scenario of Figure C.23 describes a sequence of primitives in the case when a TP-DEFERRED-END-DIALOGUE request is cancelled by a TP-ROLLBACK request during the termination phase of a provider-supported transaction. The dialogue is not terminated.

C.3.7 Dialogue establishment scenarios

C.3.7.1 Rejection of the dialogue establishment request

The scenarios of Figures C.24, C.25, and C.26 describe a sequence of primitives in the case when dialogue establishment is requested with a TP-BEGIN-DIALOGUE request, with either Confirmation = "negative" or Confirmation = "always", but is rejected by the recipient.

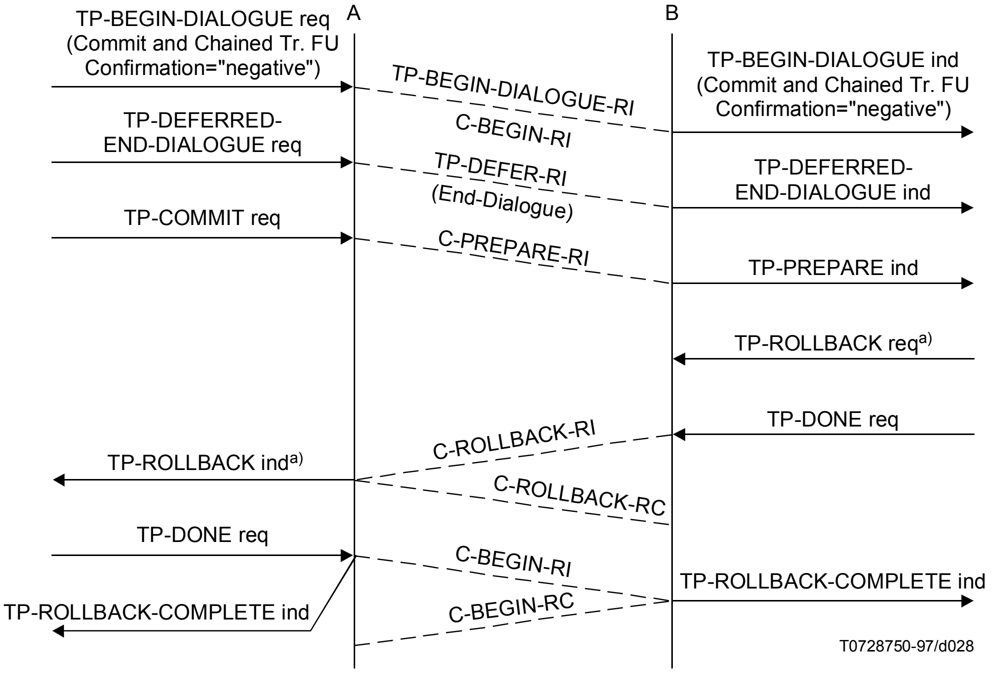
C.3.7.2 TP-BEGIN-DIALOGUE (Confirmation = "negative") scenarios

The following scenarios apply to provider-supported transaction and describe various dialogue establishment sequences using the TP-BEGIN-DIALOGUE (Confirmation = "negative") service related to aborting, rejecting and failure cases.

C.3.7.3 Aborting the dialogue establishment by the requestor

The scenario of Figure C.27 describes a sequence of primitives in the case when dialogue establishment is requested with a TP-BEGIN-DIALOGUE (Confirmation = "negative") request and is then aborted by the requestor with a TP-U-ABORT request.

When a TP-U-ABORT indication is received following a TP-BEGIN-DIALOGUE (Confirmation = "negative") indication, the TP-U-ABORT indication will carry the Rollback parameter set to "true" and the recipient TPSUI must issue a TP-DONE request to complete the rollback action.



[a)] Any other rollback-initiating service primitive would also cancel the effects of the TP-DEFERRED-END-DIALOGUE service.

NOTE – TP-BEGIN-DIALOGUE-RC and C-BEGIN-RC are not required in this case C-ROLLBACK-RI serves as the confirmation instead.

Figure C.23/X.862 – Effects of a TP-DEFERRED-END-DIALOGUE cancelled by a rollback, after TP-COMMIT request

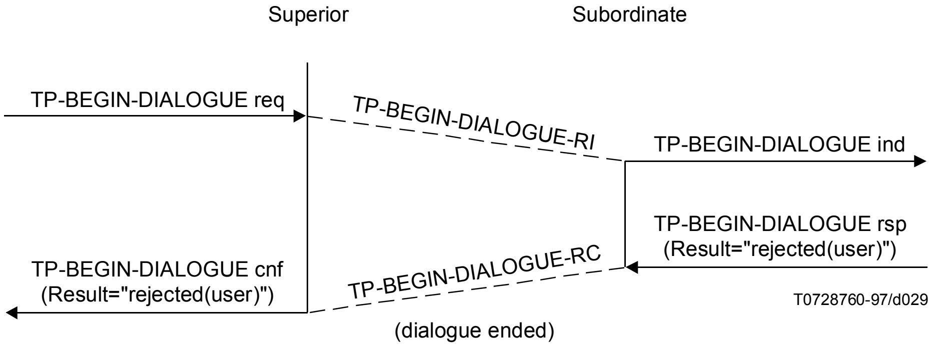


Figure C.24/X.862 – Rejection of a TP-BEGIN-DIALOGUE – Coordination level "none"

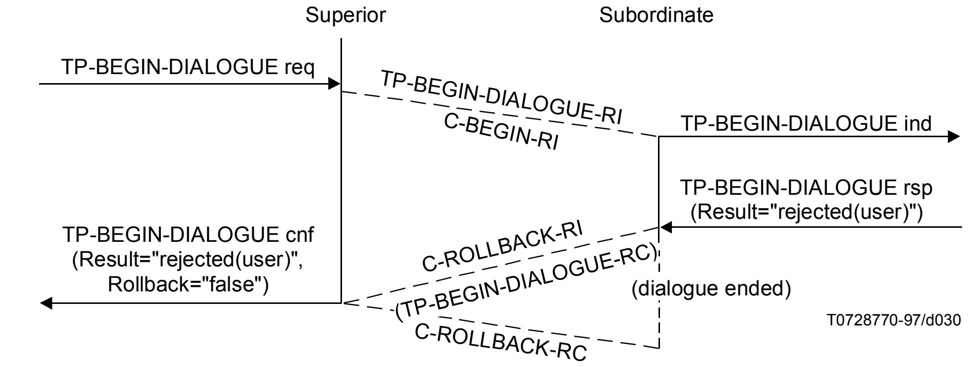


Figure C.25/X.862 – Rejection of a TP-BEGIN-DIALOGUE – Coordination level "commitment"

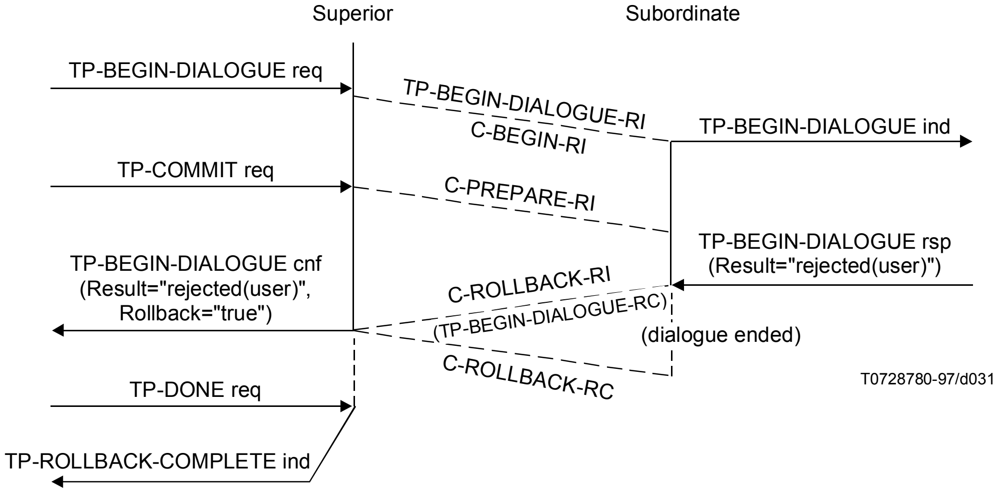


Figure C.26/X.862 – Rejection of a TP-BEGIN-DIALOGUE – Coordination level "commitment" (TP-COMMIT request issued)

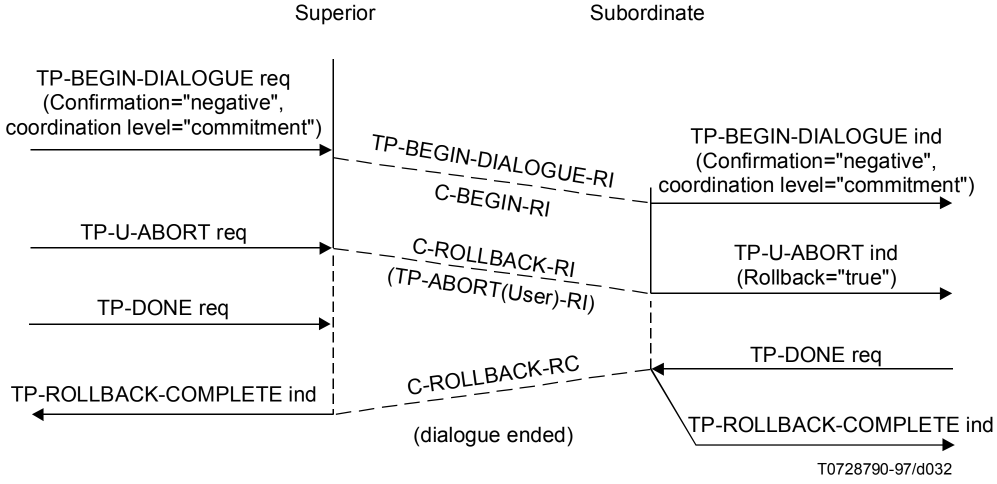


Figure C.27/X.862 – Abort of a dialogue establishment by the requestor (Confirmation = "negative")

C.3.7.4 Dialogue establishment followed by rollback from the requestor

The following scenarios describe two sequences of primitives for the cases when dialogue establishment is attempted with a TP-BEGIN-DIALOGUE (Confirmation = "negative") request and then rolled back by the requestor with a TP-ROLLBACK request.

As shown in Figure C.28, when a TP-ROLLBACK indication is received following the receipt of a TP-BEGIN-DIALOGUE (Confirmation = "negative") indication and prior to issuing any other services on the specific dialogue, a negative response may be issued with a TP-BEGIN-DIALOGUE (Result = "rejected(user)") response.

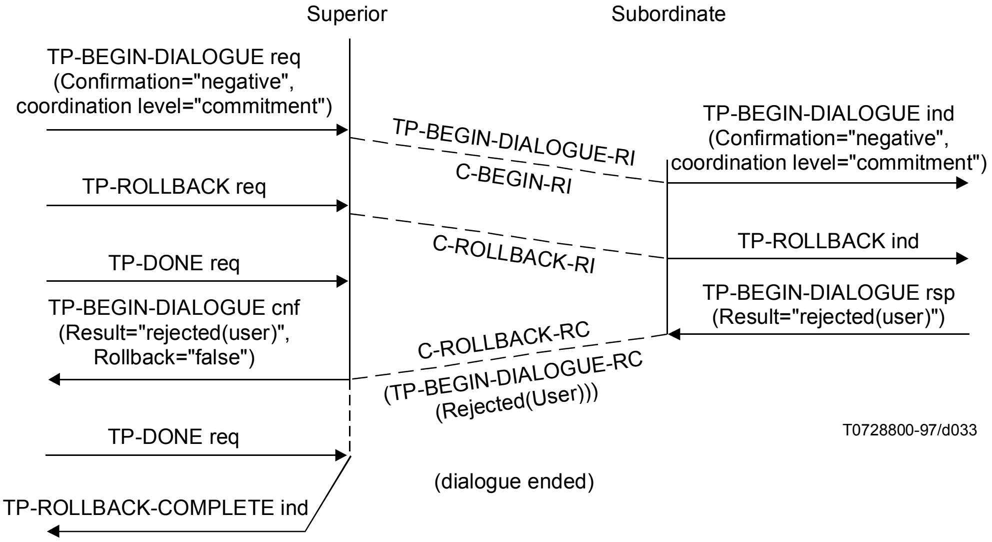

Superior
Subordinate
TP-BEGIN-DIALOGUE req (Confirmation="negative", coordination level="commitment")
TP-BEGIN-DIALOGUE-RI
C-BEGIN-RI
TP-BEGIN-DIALOGUE ind (Confirmation="negative", coordination level="commitment")
TP-ROLLBACK req
C-ROLLBACK-RI
TP-ROLLBACK ind
TP-DONE req
TP-BEGIN-DIALOGUE rsp (Result="rejected(user)")
TP-BEGIN-DIALOGUE cnf (Result="rejected(user)", Rollback="false")
C-ROLLBACK-RC
(TP-BEGIN-DIALOGUE-RC (Rejected(User)))
TP-DONE req
T0728800-97/d033
(dialogue ended)
TP-ROLLBACK-COMPLETE ind

Figure C.28/X.862 – Dialogue establishment followed by rollback from the requestor – Recipient rejects the dialogue

Otherwise, the recipient must issue a TP-DONE request to complete the rollback action, as shown in Figure C.29.

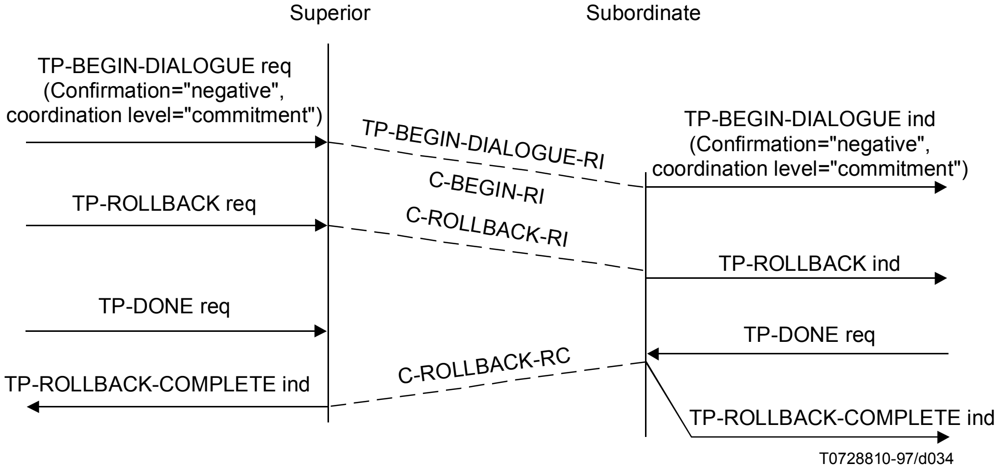


Figure C.29/X.862 – Dialogue establishment followed by rollback from the requestor – Recipient accepts the dialogue

C.3.7.5 Failure after Dialogue Establishment

The scenario of Figure C.30 describes a sequence of primitives in the case when dialogue establishment is attempted with a TP-BEGIN-DIALOGUE (Confirmation = "negative") request, followed by a failure.

In case of a dialogue failure, a TP-P-ABORT indication with the Rollback parameter set to "true" will be issued to both TPSUIs. When the TP-P-ABORT indication is issued to the superior TPSUI before the TP-BEGIN-DIALOGUE confirm has been received, the TP-P-ABORT indication will have the Rollback parameter set to "true" because it is not known whether the subordinate accepted or rejected the dialogue initialization request.

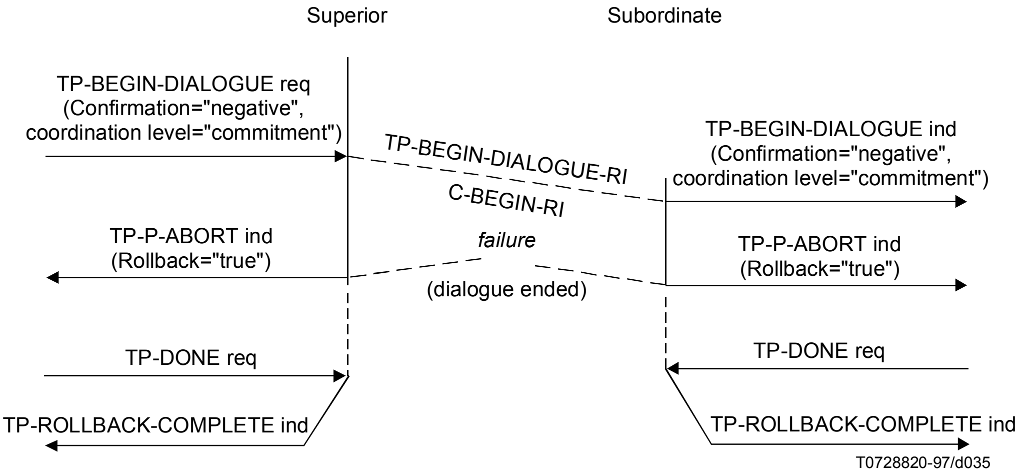

Superior
Subordinate
TP-BEGIN-DIALOGUE req (Confirmation="negative", coordination level="commitment")
TP-BEGIN-DIALOGUE-RI
C-BEGIN-RI
TP-BEGIN-DIALOGUE ind (Confirmation="negative", coordination level="commitment")
TP-P-ABORT ind (Rollback="true")
failure
(dialogue ended)
TP-P-ABORT ind (Rollback="true")
TP-DONE req
TP-DONE req
TP-ROLLBACK-COMPLETE ind
TP-ROLLBACK-COMPLETE ind
T0728820-97/d035

Figure C.30/X.862 – Dialogue establishment (Confirmation = "negative") followed by dialogue failure – Recipient does not reject the dialogue

C.3.7.6 Dialogue establishment rejected after TP-COMMIT request

Figure C.31 shows how dialogue establishment rejection – received after a TP-COMMIT request – signals that the transaction is being rolled back.

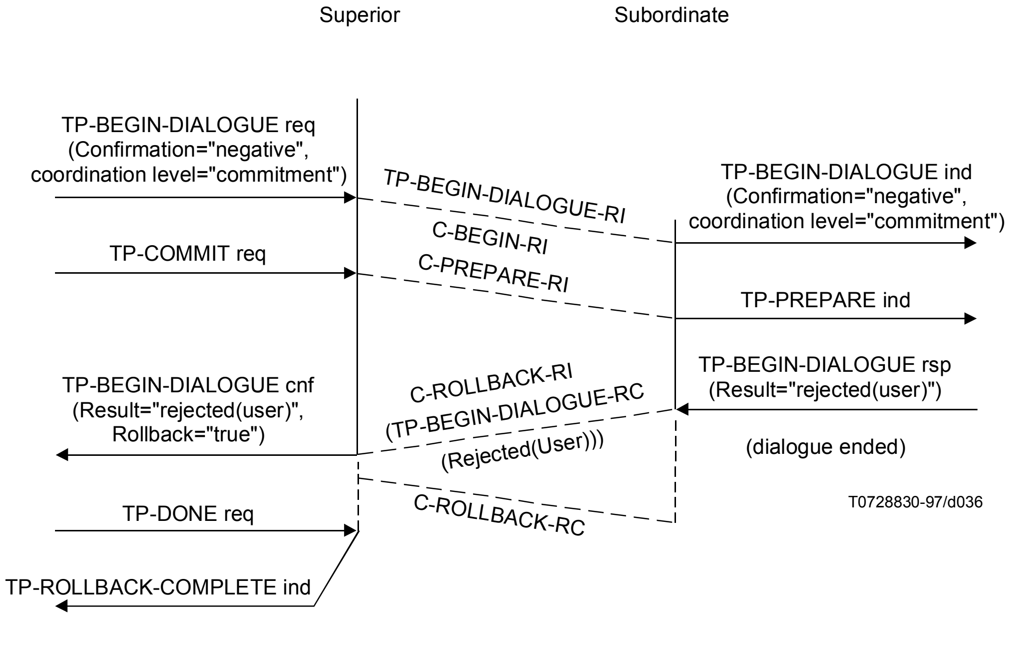

Superior
Subordinate
TP-BEGIN-DIALOGUE req (Confirmation="negative", coordination level="commitment")
TP-BEGIN-DIALOGUE-RI
C-BEGIN-RI
TP-BEGIN-DIALOGUE ind (Confirmation="negative", coordination level="commitment")
TP-COMMIT req
C-PREPARE-RI
TP-PREPARE ind
TP-BEGIN-DIALOGUE cnf (Result="rejected(user)", Rollback="true")
C-ROLLBACK-RI
(TP-BEGIN-DIALOGUE-RC
(Rejected(User)))
TP-BEGIN-DIALOGUE rsp (Result="rejected(user)")
(dialogue ended)
C-ROLLBACK-RC
TP-DONE req
TP-ROLLBACK-COMPLETE ind
T0728830-97/d036

Figure C.31/X.862 – Rejection of a Dialogue establishment request causing rollback

C.3.7.7 Dialogue establishment rejected after TP-PREPARE request

Figure C.32 shows how dialogue establishment rejection received after a TP-PREPARE request does not cause rollback of the transaction.

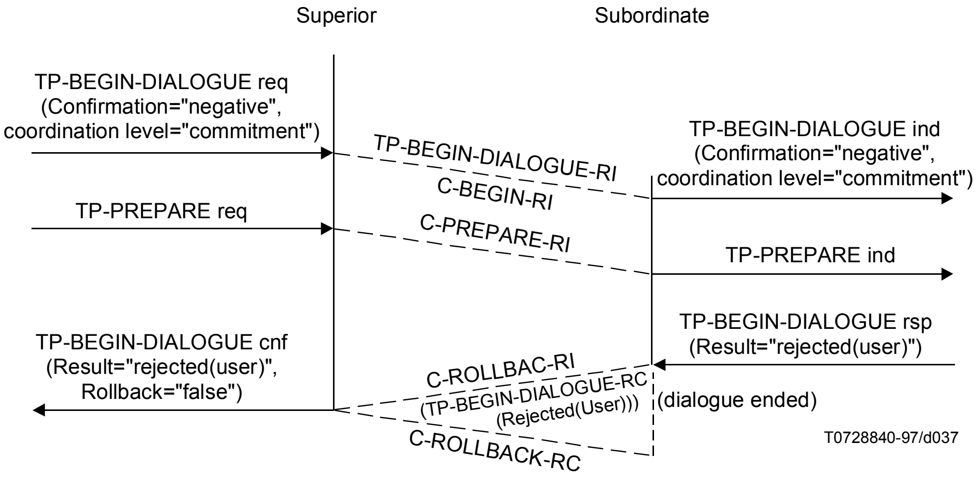


Figure C.32/X.862 – Rejection of a Dialogue establishment request after TP-PREPARE

C.3.7.8 TP-BEGIN-DIALOGUE (Confirmation = "always") scenarios

The following scenarios describe various dialogue establishment sequences using the TP-BEGIN-DIALOGUE (Confirmation = "always") service related to aborting, rejecting and failure cases.

C.3.7.9 Aborting the dialogue establishment

The scenario of Figure C.33 describes sequences of primitives for the case when dialogue establishment is attempted with a TP-BEGIN-DIALOGUE (Confirmation = "always") request and then aborted, by the requestor, with a TP-U-ABORT request.

When a TP-U-ABORT indication is received following a TP-BEGIN-DIALOGUE (Confirmation = "always") indication and before the indication has been responded to, the TP-U-ABORT indication will carry the Rollback parameter set to "false", and no further actions are required by the subordinate.

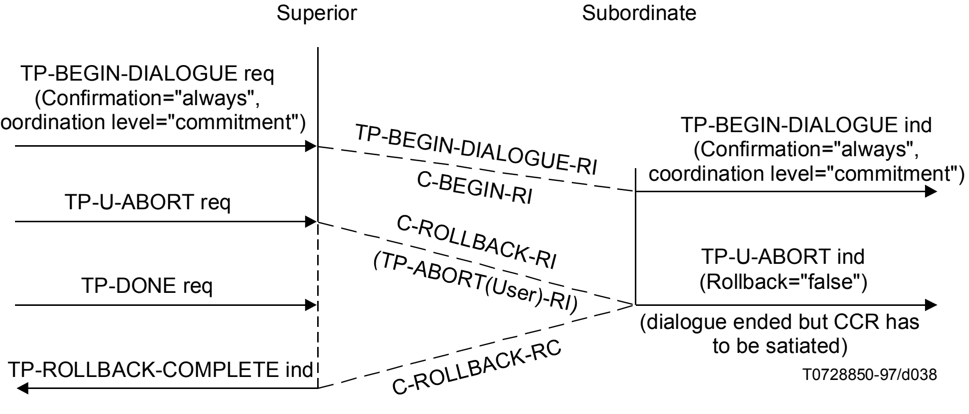


Figure C.33/X.862 – Abort of dialogue establishment by the requestor (Confirmation = "always"), before the recipient has accepted the dialogue

When a TP-U-ABORT indication is received following a TP-BEGIN-DIALOGUE (Confirmation = "always") indication but after the indication has been positively responded to, the TP-U-ABORT indication will carry the Rollback parameter set to "true", and the subordinate must complete normal rollback actions, as shown in Figure C.34.

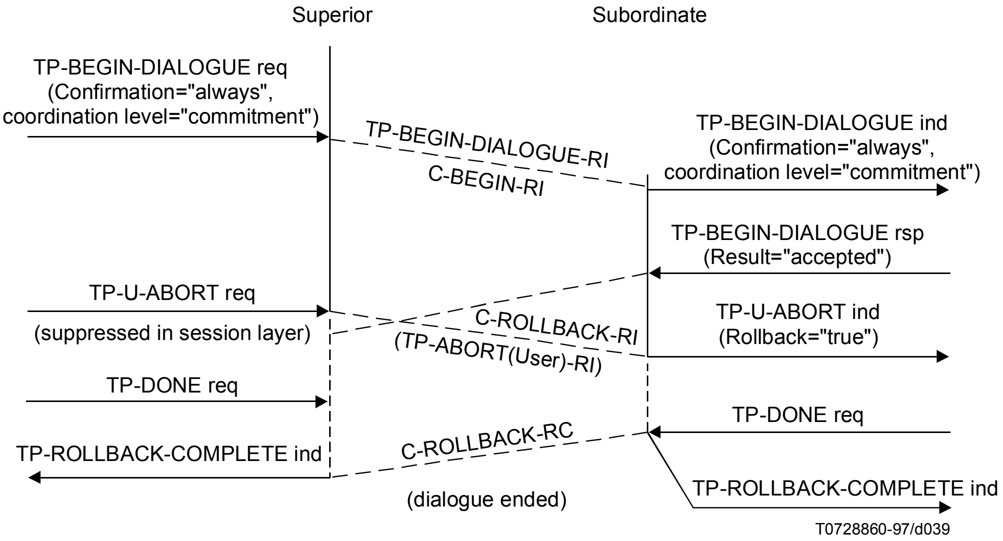

Superior
Subordinate
TP-BEGIN-DIALOGUE req (Confirmation="always", coordination level="commitment")
TP-BEGIN-DIALOGUE-RI
C-BEGIN-RI
TP-BEGIN-DIALOGUE ind (Confirmation="always", coordination level="commitment")
TP-BEGIN-DIALOGUE rsp (Result="accepted")
TP-U-ABORT req
(suppressed in session layer)
C-ROLLBACK-RI
(TP-ABORT(User)-RI)
TP-U-ABORT ind (Rollback="true")
TP-DONE req
TP-DONE req
TP-ROLLBACK-COMPLETE ind
C-ROLLBACK-RC
(dialogue ended)
TP-ROLLBACK-COMPLETE ind
T0728860-97/d039

Figure C.34/X.862 – Abort of a dialogue establishment (Confirmation = "always") by the requestor, after the recipient has accepted the dialogue

The scenario in Figure C.35 shows that the recipient of a TP-BEGIN-DIALOGUE (Confirmation = "always") indication, if it wants to abort the dialogue, shall issue the TP-BEGIN-DIALOGUE response before the TP-U-ABORT request can be issued.

C.3.7.10 Dialogue establishment followed by rollback from the requestor

These scenarios describe two sequences of primitives in the cases when dialogue establishment is attempted with a TP-BEGIN-DIALOGUE (Confirmation = "always") request and then rolled back by the requestor with a TP-ROLLBACK request.

When a TP-ROLLBACK indication is received before a TP-BEGIN-DIALOGUE (Confirmation = "always") indication has been responded to, the TP-BEGIN-DIALOGUE response must be issued before the TP-ROLLBACK indication is acted on; a negative response to the TP-BEGIN-DIALOGUE (confirmed) indication completes the subordinate’s rollback action, as shown in Figure C.36.

The recipient TPSUI may also issue a positive response to the TP-BEGIN-DIALOGUE (Confirmation = "always") indication, followed by a TP-DONE request, as shown in Figure C.37.

C.3.7.11 Failure during dialogue establishment

The scenario of Figure C.38 describes a sequence of primitives in the case when dialogue establishment is attempted with a TP-BEGIN-DIALOGUE (Confirmation = "always") request, followed by a failure. In case of a dialogue failure, a TP-P-ABORT indication will be issued to the superior with the Rollback parameter set to "true" and to the subordinate with the Rollback parameter set to "false".

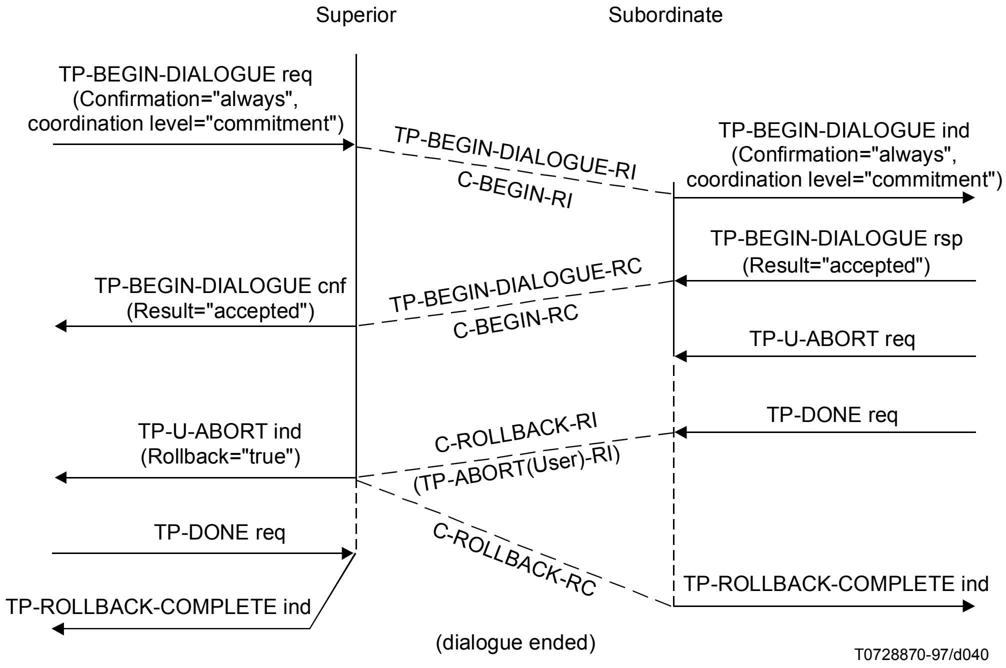

Superior
Subordinate
TP-BEGIN-DIALOGUE req (Confirmation="always", coordination level="commitment")
TP-BEGIN-DIALOGUE-RI
C-BEGIN-RI
TP-BEGIN-DIALOGUE ind (Confirmation="always", coordination level="commitment")
TP-BEGIN-DIALOGUE rsp (Result="accepted")
TP-BEGIN-DIALOGUE cnf (Result="accepted")
TP-BEGIN-DIALOGUE-RC
C-BEGIN-RC
TP-U-ABORT req
TP-DONE req
TP-U-ABORT ind (Rollback="true")
C-ROLLBACK-RI
(TP-ABORT(User)-RI)
TP-DONE req
C-ROLLBACK-RC
TP-ROLLBACK-COMPLETE ind
TP-ROLLBACK-COMPLETE ind
(dialogue ended)
T0728870-97/d040

Figure C.35/X.862 – Abort of a dialogue establishment (Confirmation = "always") by the recipient, after the recipient has accepted the dialogue

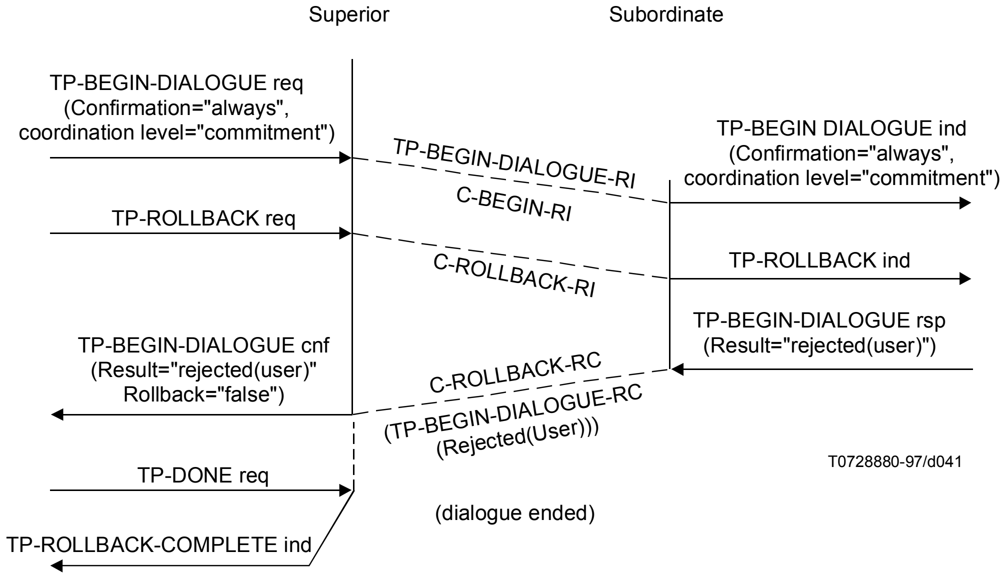

Superior
Subordinate
TP-BEGIN-DIALOGUE req (Confirmation="always", coordination level="commitment")
TP-BEGIN-DIALOGUE-RI
C-BEGIN-RI
TP-BEGIN DIALOGUE ind (Confirmation="always", coordination level="commitment")
TP-ROLLBACK req
C-ROLLBACK-RI
TP-ROLLBACK ind
TP-BEGIN-DIALOGUE rsp (Result="rejected(user)")
TP-BEGIN-DIALOGUE cnf (Result="rejected(user)" Rollback="false")
C-ROLLBACK-RC
(TP-BEGIN-DIALOGUE-RC (Rejected(User)))
TP-DONE req
TP-ROLLBACK-COMPLETE ind
(dialogue ended)
T0728880-97/d041

Figure C.36/X.862 – Dialogue establishment followed by rollback from the requestor – Recipient rejects the dialogue

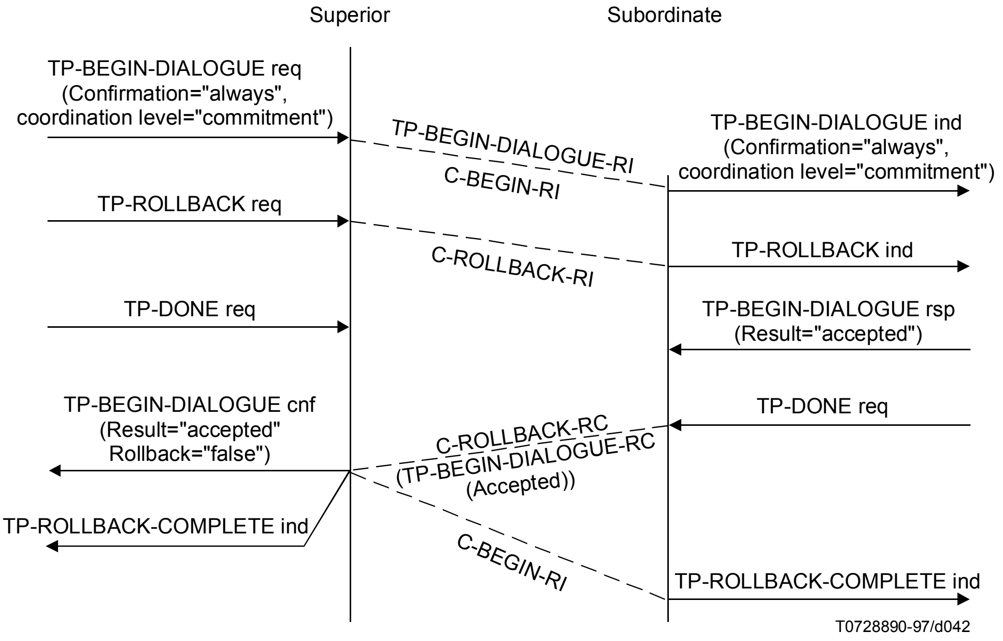

Superior
Subordinate
TP-BEGIN-DIALOGUE req (Confirmation="always", coordination level="commitment")
TP-BEGIN-DIALOGUE-RI
C-BEGIN-RI
TP-BEGIN-DIALOGUE ind (Confirmation="always", coordination level="commitment")
TP-ROLLBACK req
C-ROLLBACK-RI
TP-ROLLBACK ind
TP-DONE req
TP-BEGIN-DIALOGUE rsp (Result="accepted")
TP-BEGIN-DIALOGUE cnf (Result="accepted" Rollback="false")
C-ROLLBACK-RC
(TP-BEGIN-DIALOGUE-RC (Accepted))
TP-DONE req
TP-ROLLBACK-COMPLETE ind
C-BEGIN-RI
TP-ROLLBACK-COMPLETE ind
T0728890-97/d042

Figure C.37/X.862 – Dialogue establishment followed by rollback from the requestor – Recipient accepts the dialogue and has to complete the rollback procedure (chained transactions)

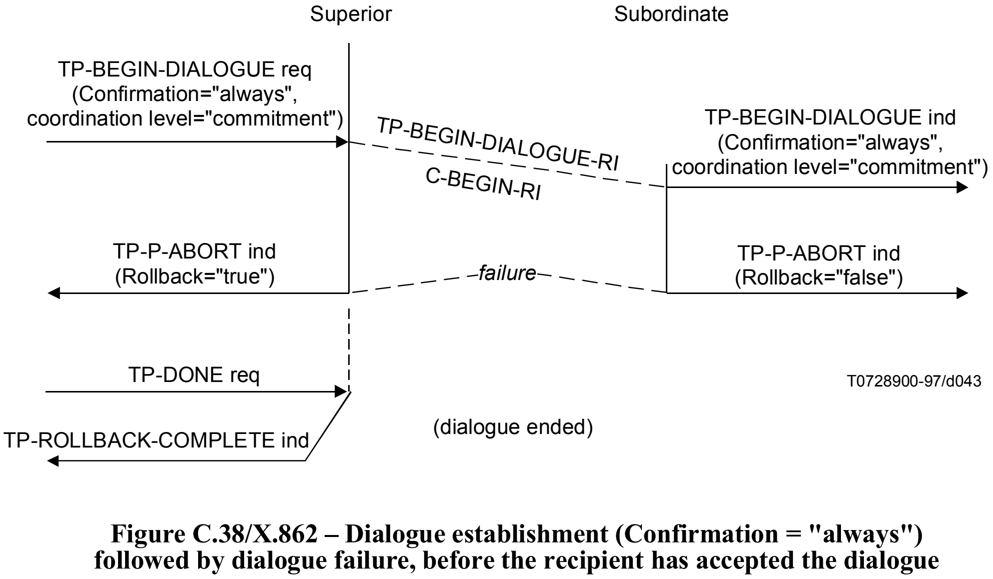

Superior
Subordinate
TP-BEGIN-DIALOGUE req (Confirmation="always", coordination level="commitment")
TP-BEGIN-DIALOGUE-RI
C-BEGIN-RI
TP-BEGIN-DIALOGUE ind (Confirmation="always", coordination level="commitment")
TP-P-ABORT ind (Rollback="true")
failure
TP-P-ABORT ind (Rollback="false")
TP-DONE req
T0728900-97/d043
TP-ROLLBACK-COMPLETE ind
(dialogue ended)

Figure C.38/X.862 – Dialogue establishment (Confirmation = "always") followed by dialogue failure, before the recipient has accepted the dialogue

However, if the subordinate has already issued a positive TP-BEGIN-DIALOGUE response, the TP-P-ABORT will carry the Rollback parameter set to "true" and the subordinate shall complete the rollback actions in the usual manner, as shown in Figure C.39.

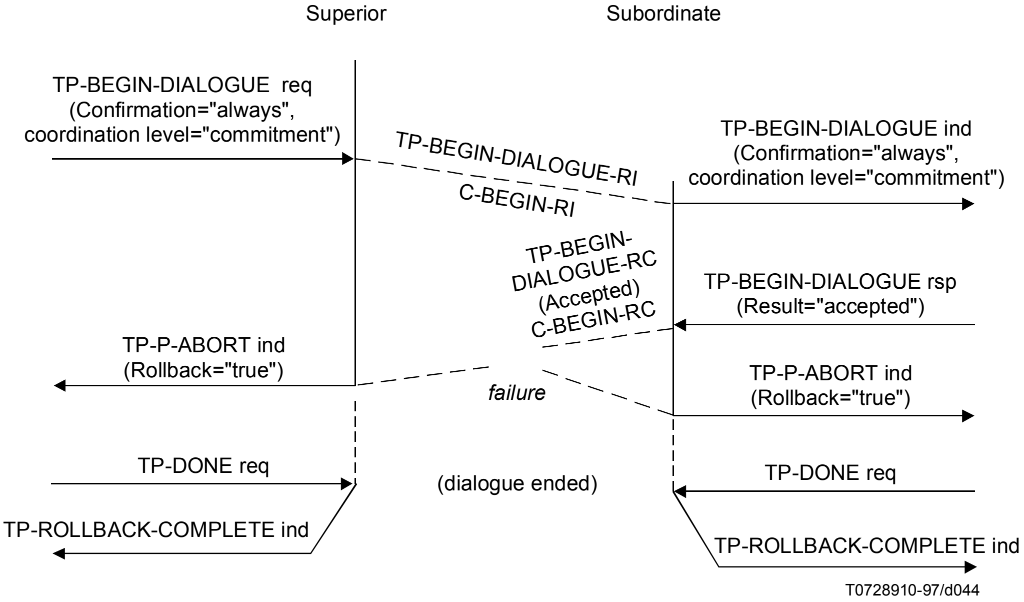


Figure C.39/X.862 – Dialogue establishment (Confirmation = "always") followed by failure, after the recipient has accepted the dialogue – The recipient has to complete the rollback procedure

C.3.7.12 Dialogue establishment with rollback and abort

These scenarios cover the case when dialogue establishment is attempted with a TP-BEGIN-DIALOGUE (Confirmation = "always") request and then the transaction is rolled back by the requestor with a TP-ROLLBACK request.

Before a TP-BEGIN-DIALOGUE confirm is received, the dialogue is aborted by either a TP-P-ABORT indication (see Figure C.40) or a TP-U-ABORT request (see Figures C.41 and C.42).

When a TP-P-ABORT indication is received, subsequent to the TP-ROLLBACK indication, this ends the termination phase of the transaction as shown in Figure C.40.

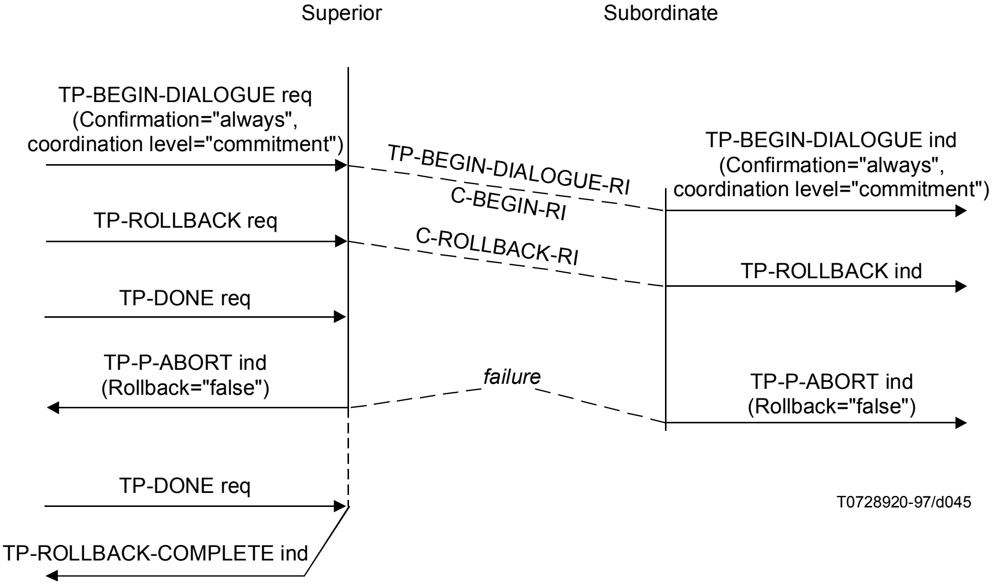


Figure C.40/X.862 – Dialogue establishment followed by rollback from the requestor and a subsequent dialogue abort

The scenario of Figure C.41 shows the case where the superior issues a TP-U-ABORT request for the dialogue. Here the TP-U-ABORT indication is withheld until the positive dialogue establishment confirmation is received.

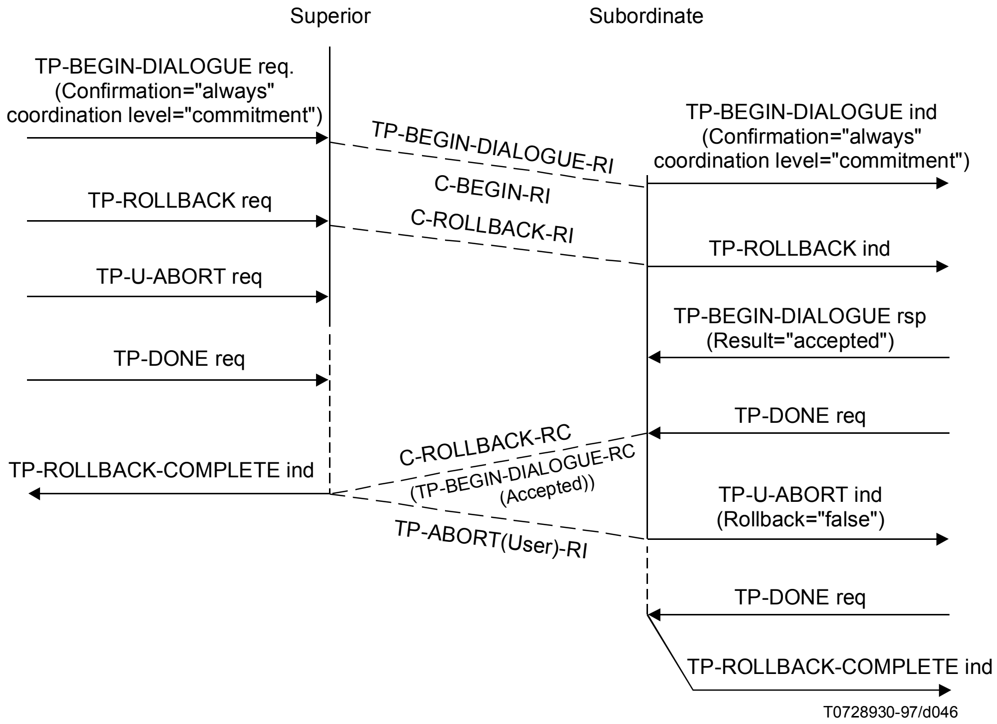


Figure C.41/X.862 – Dialogue establishment followed by rollback by the requestor and a subsequent collision of TP-U-ABORT request and TP-BEGIN-DIALOGUE response (Result = "accepted") – The Chained Transactions functional unit is selected

The scenario of Figure C.42 is similar to that shown on Figure C.41 except that the subordinate rejects the dialogue after receiving the TP-ROLLBACK indication.

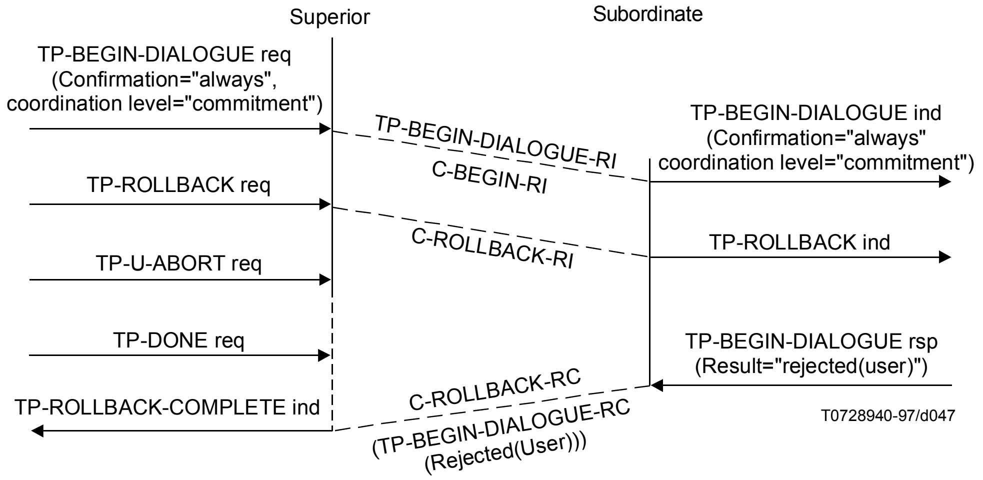


Figure C.42/X.862 – Dialogue establishment followed by rollback by the requestor and a subsequent collision of TP-U-ABORT request and TP-BEGIN-DIALOGUE response (Result = "rejected(user)")

C.4 Scenarios with a single dialogue (failure cases)

C.4.1 TP-P-ABORT

These scenarios apply to dialogues with a coordination level of "commitment" with either chained and unchained transactions unless otherwise specified.

C.4.1.1 TP-P-ABORT during active phase

The scenario of Figure C.43 describes a sequence of primitives in the case when a provider-supported transaction is aborted by a communications failure of the dialogue during the active phase.

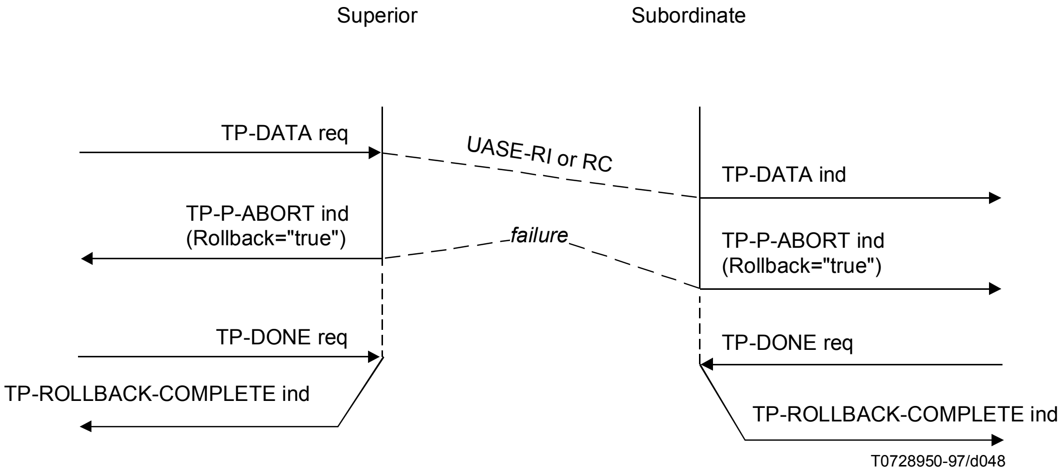


Figure C.43/X.862 – Failure for a provider-supported transaction in the active phase

C.4.1.2 TP-P-ABORT during the first phase of commitment

The scenario of Figure C.44 describes a sequence of primitives in the case when a provider-supported transaction is aborted by a communications failure of the dialogue during the first phase of commitment.

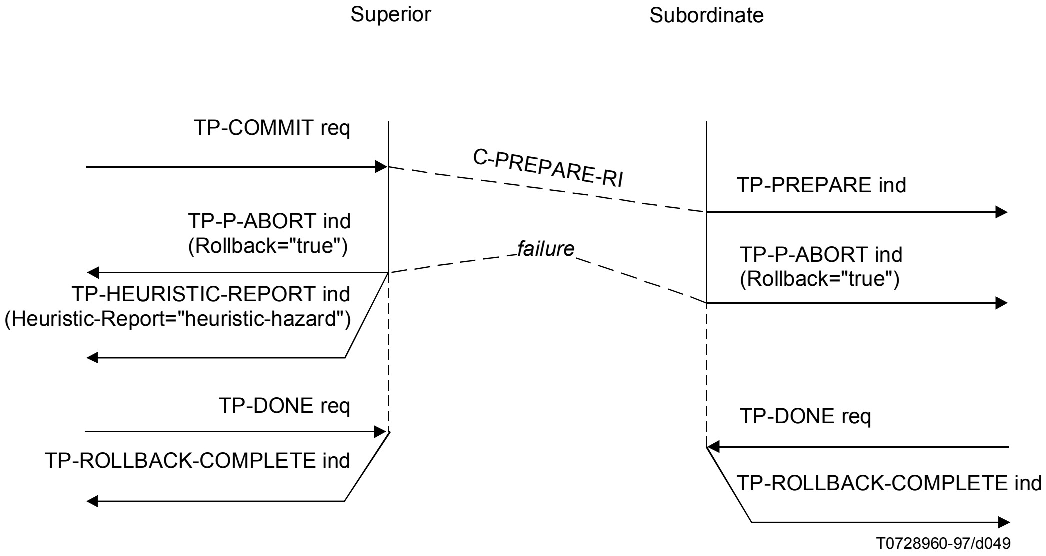


Figure C.44/X.862 – Failure for a provider-supported transaction during the first phase of commitment

C.4.1.3 TP-P-ABORT during the second phase of commitment

The scenarios of Figures C.45 to C.47 describe sequences of primitives in the cases when a dialogue is aborted during the second phase of commitment. The three scenarios illustrate the effect that the precise timing of a failure has during commitment.

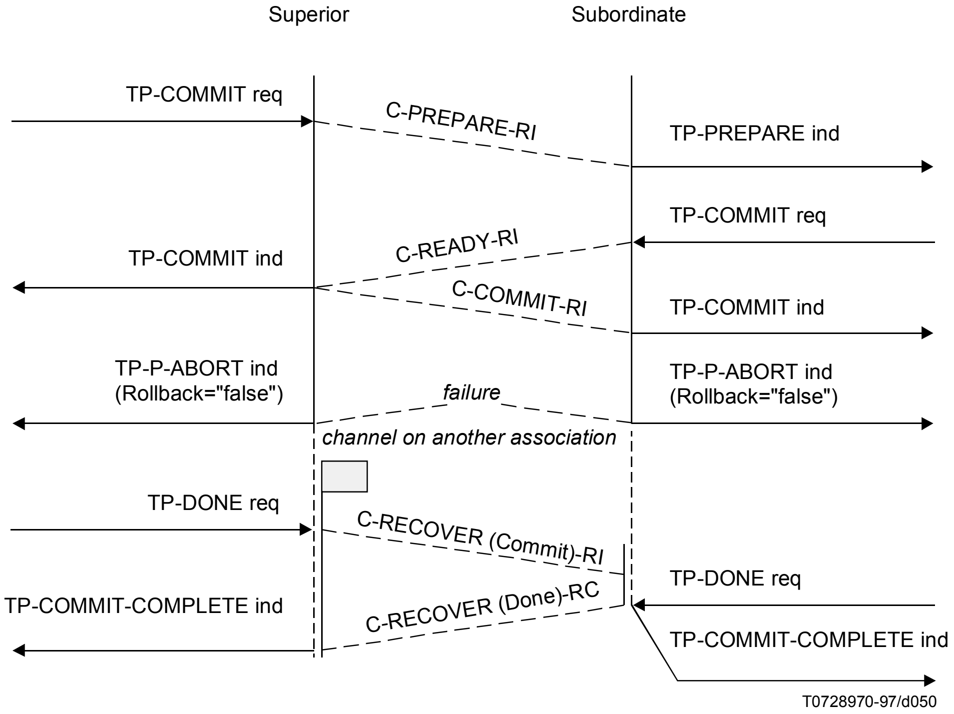


Figure C.45/X.862 – Failure for a provider-supported transaction during the second phase of commitment

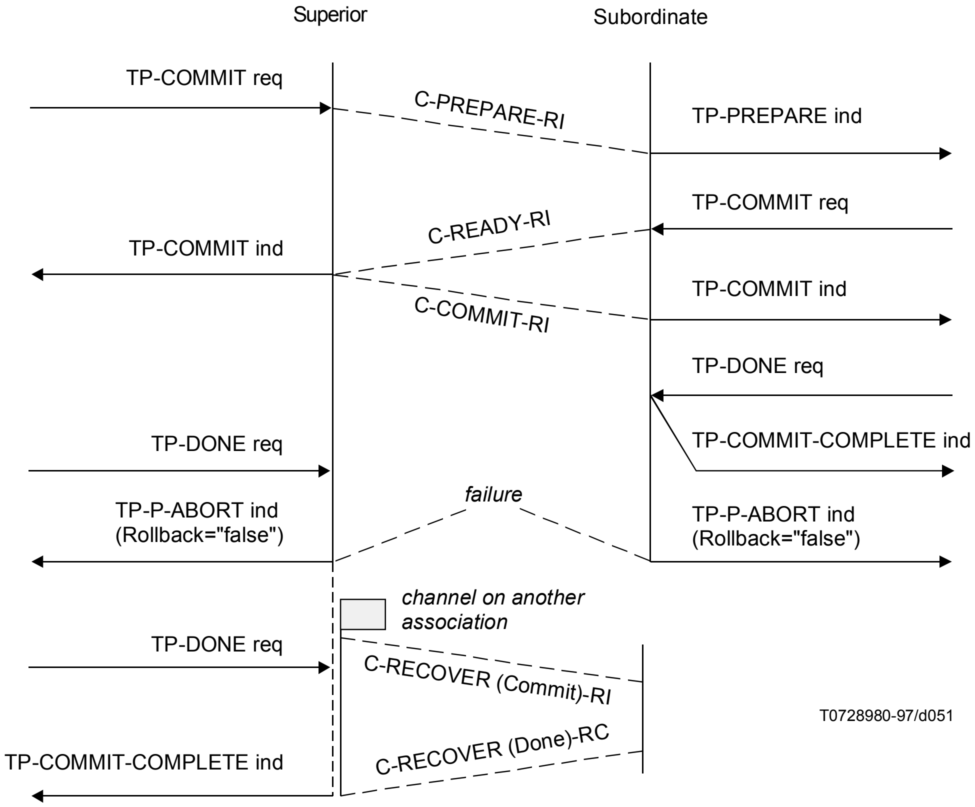


Figure C.46/X.862 – Failure for a provider-supported transaction during the second phase of commitment when the Unchained Transactions functional unit is selected

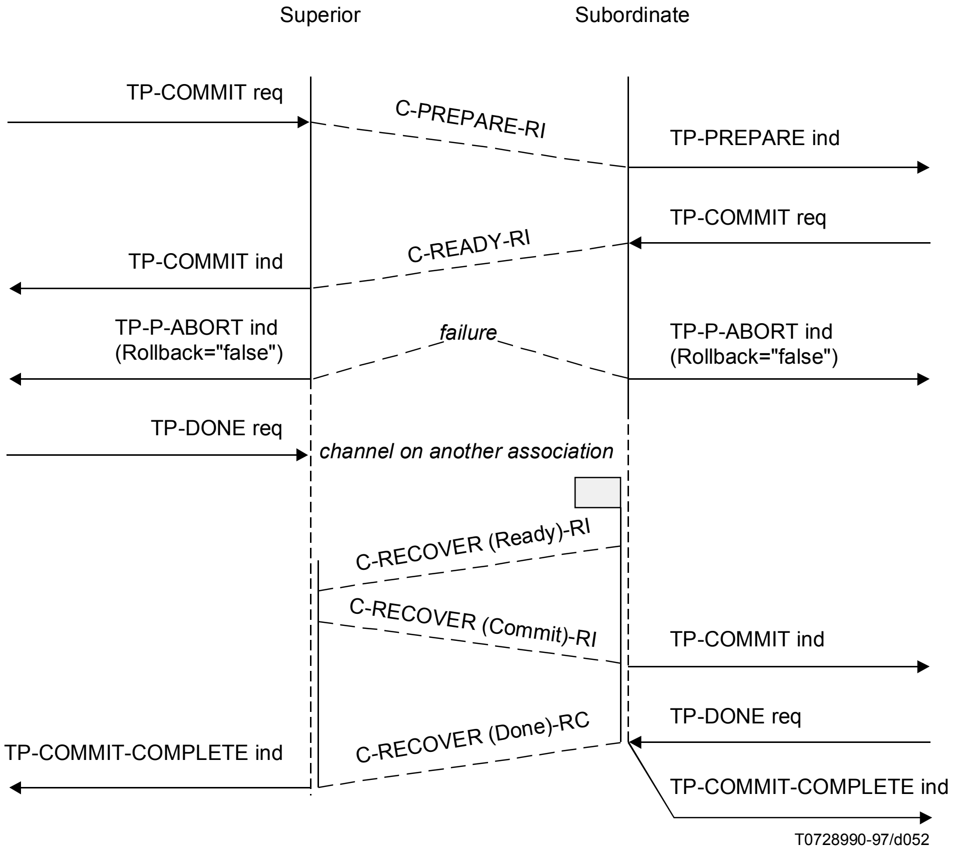


Figure C.47/X.862 – Failure for a provider-supported transaction during the second phase of commitment

C.5 Collision scenarios on a single dialogue

C.5.1 Collisions of TP-U-ERROR with TP-COMMIT

The scenario of Figure C.48 shows that, if a TP-U-ERROR and a TP-COMMIT collide, the TP-U-ERROR indication is suppressed and a rollback procedure is triggered.

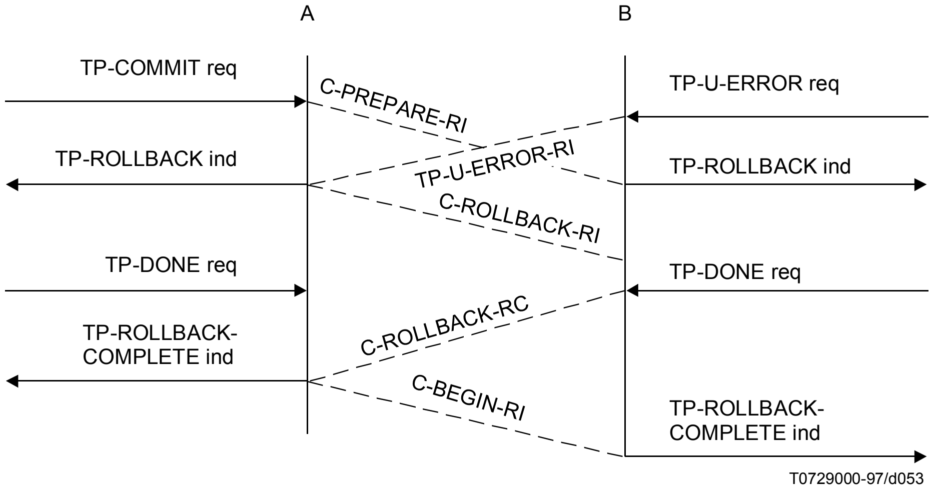


Figure C.48/X.862 – Collision of TP-COMMIT and TP-U-ERROR

C.5.2 Collisions with TP-U-ERROR in Polarized Control

After a TP-U-ERROR request issued when the TPSUI does not have the control of the dialogue, any TP-U-ERROR indications and TP-DATA indication are suppressed according to the scenario of Figure C.49.

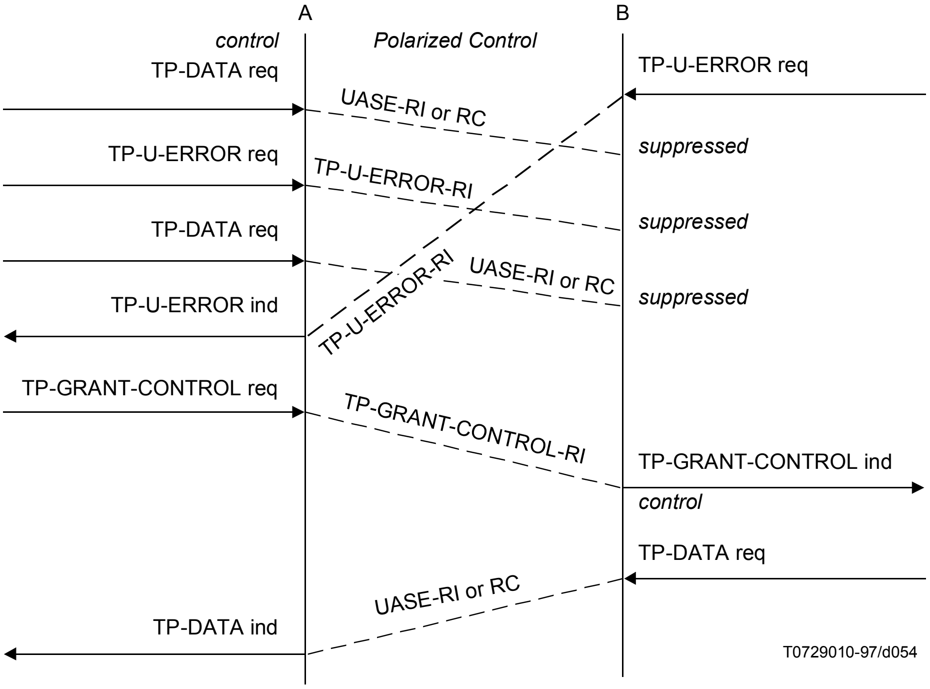


Figure C.49/X.862 – Suppression effects of TP-U-ERROR

Due to the transit delay of the Grant Control service, it may happen that two TP-U-ERROR requests sent by TPSUIs not having control collide. This collision is shown by the scenario of Figure C.50.

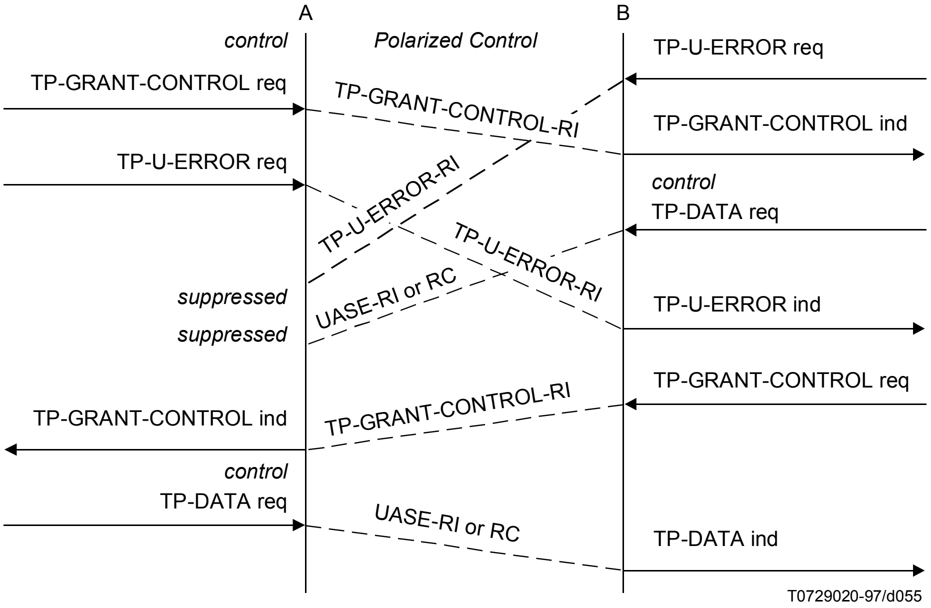


Figure C.50/X.862 – Collisions of two TP-U-ERRORs issued without having the control

C.5.3 Collisions with TP-U-ERROR in Shared Control

The scenario of Figure C.51 shows the suppression effect of TP-U-ERROR in Shared Control.

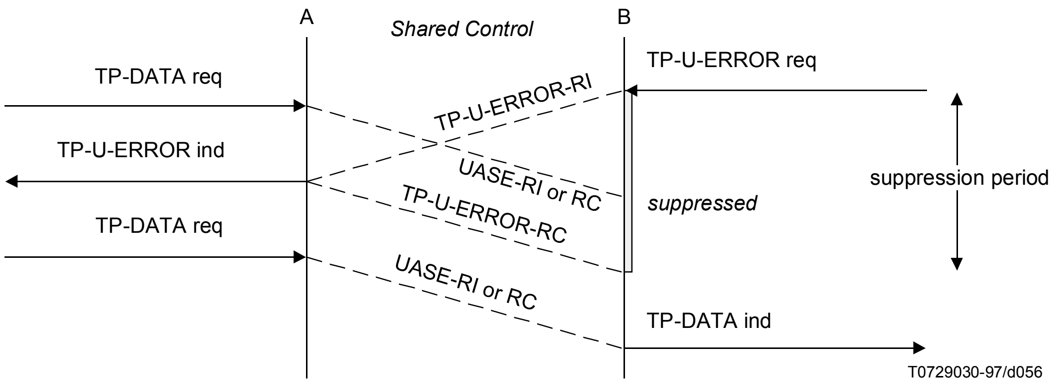


Figure C.51/X.862 – Suppression effect of TP-U-ERROR in Shared Control

If a collision occurs between two TP-U-ERROR requests, the suppression is shown in Figure C.52.

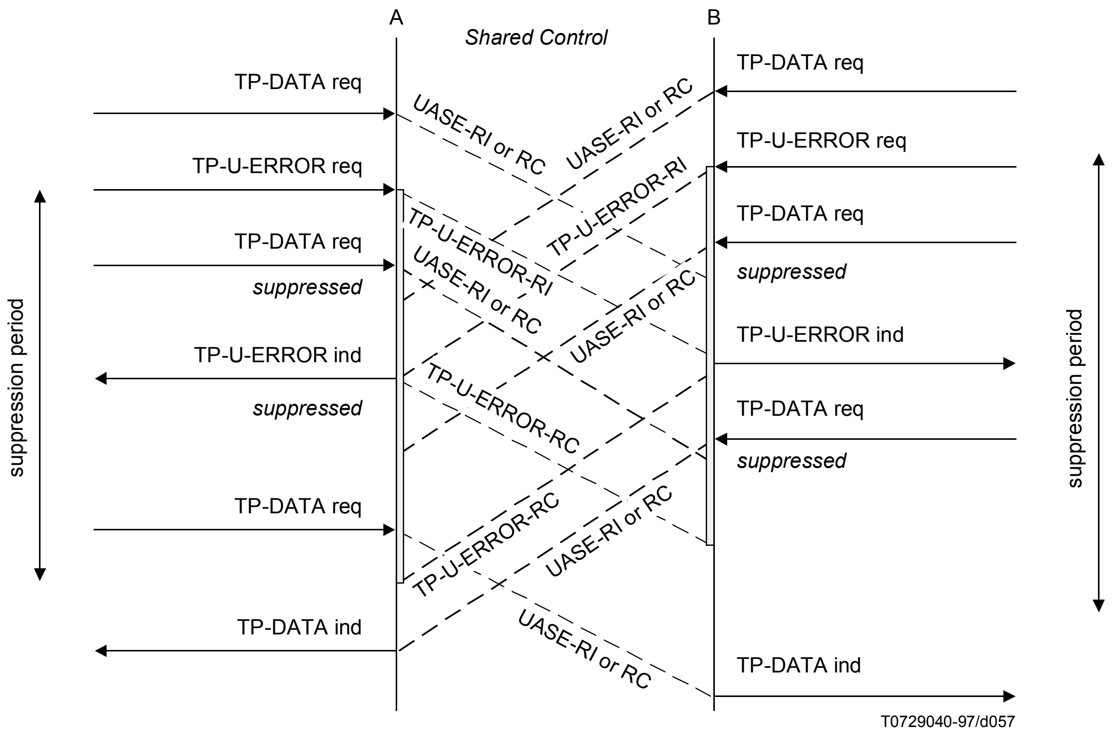


Figure C.52/X.862 – Collision of TP-U-ERRORs in Shared Control

The scenarios of Figures C.53 and C.54 shows that if TP-U-ERROR request is issued between a TP-HANDSHAKE request and the corresponding confirmation, the requestor of TP-U-ERROR request cannot issue a TP-DATA request until it has received the handshake confirmation.

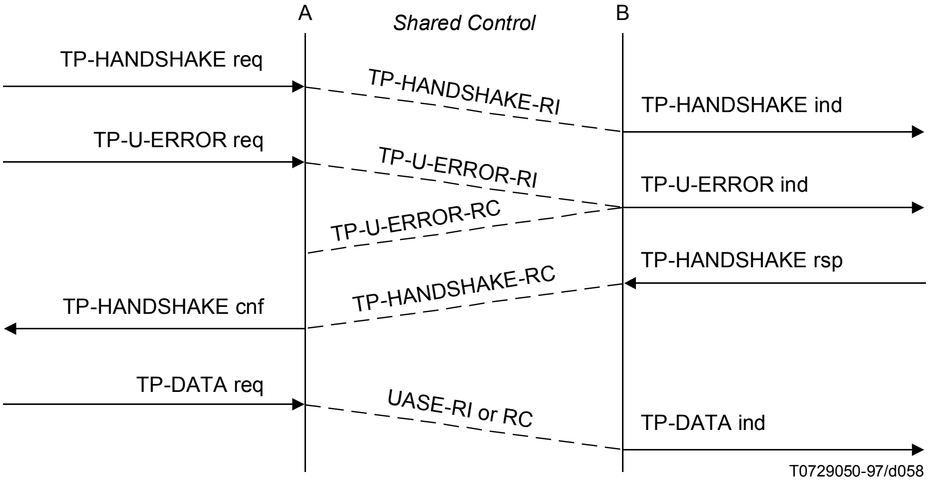


Figure C.53/X.862 – TP-U-ERROR during an outstanding handshake request

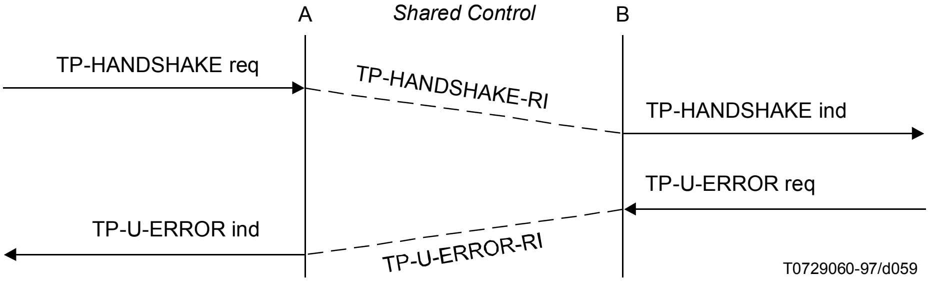


Figure C.54/X.862 – TP-U-ERROR during an outstanding handshake indication

The scenario in Figure C.55 shows that, in Shared Control, a TP-U-ERROR request is not confirmed when colliding with a TP-END-DIALOGUE request (Confirmation = "true"). It would be the same for the TP-U-ERROR request / TP-HANDSHAKE request collision.

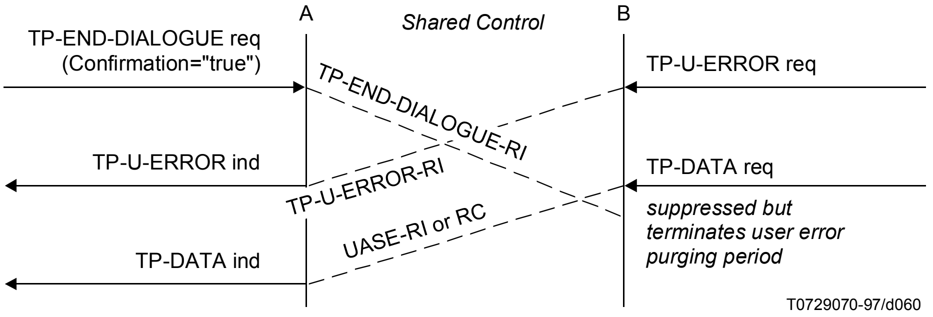


Figure C.55/X.862 – Collision of a TP-U-ERROR request with a TP-END-DIALOGUE indication

The scenario in Figure C.56 shows a case when, in Shared Control, two TP-U-ERROR requests collide with a TP-HANDSHAKE request; only the first TP-U-ERROR-RI is not confirmed.

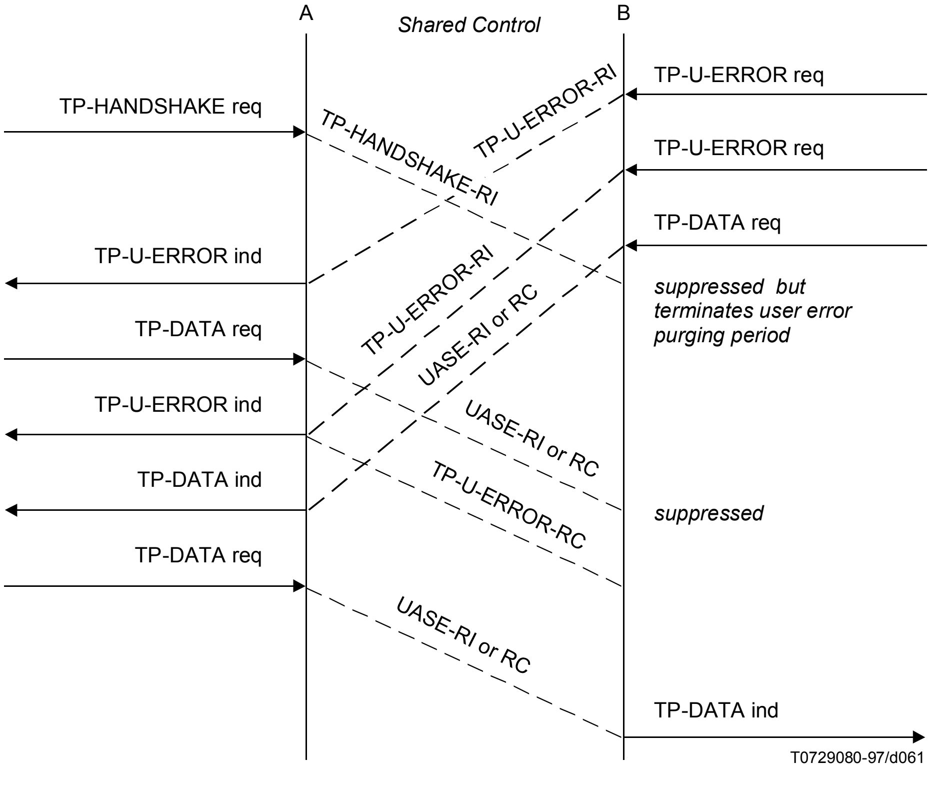


Figure C.56/X.862 – Collision of two TP-U-ERROR requests with a TP-HANDSHAKE indication in Shared Control

The scenario in Figure C.57 shows that TP-U-ERROR-RCs are not sent when a dialogue establishment indication is outstanding; they are sent after the TP-BEGIN-DIALOGUE-RC, and after a C-BEGIN-RC (if any).

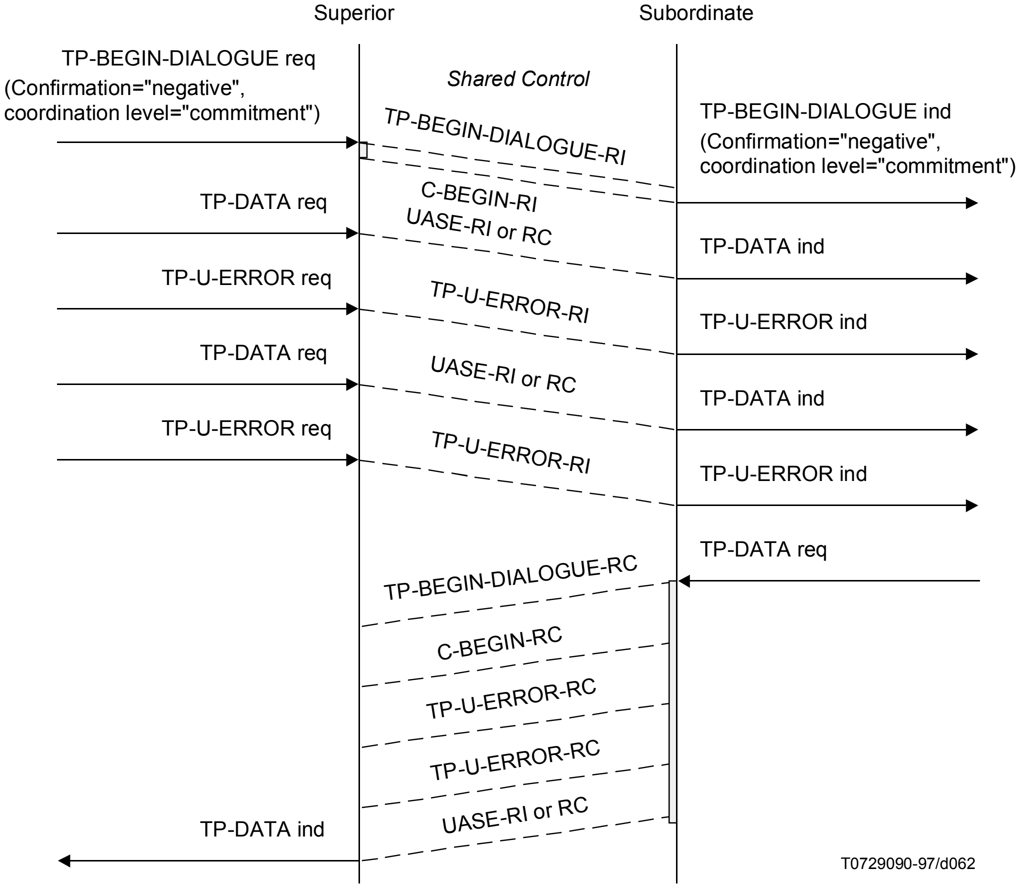


Figure C.57/X.862 – Queuing TP-U-ERROR-RCs before TP-BEGIN-DIALOGUE-RC is sent in Shared Control

The scenario in Figure C.58 shows that two TP-U-ERROR requests, sent in response to two colliding TP-HANDSHAKE requests, are not confirmed.

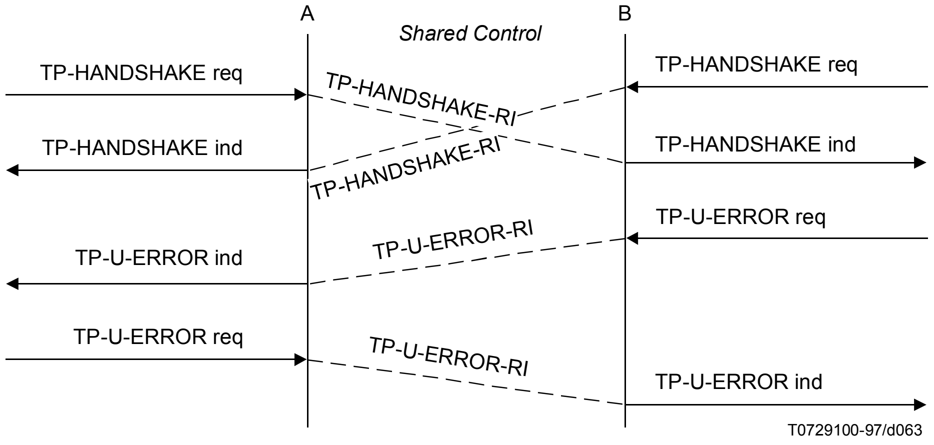


Figure C.58/X.862 – Two TP-U-ERROR requests after a TP-HANDSHAKE collision in Shared Control

C.5.4 TP-END-DIALOGUE Collision Scenarios

C.5.4.1 Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-END-DIALOGUE (Confirmation = "false")

The scenario of Figure C.59 describes a sequence of primitives in the case when a TP-END-DIALOGUE (Confirmation = "true") request collides with a TP-END-DIALOGUE (Confirmation = "false") request in Shared Control.

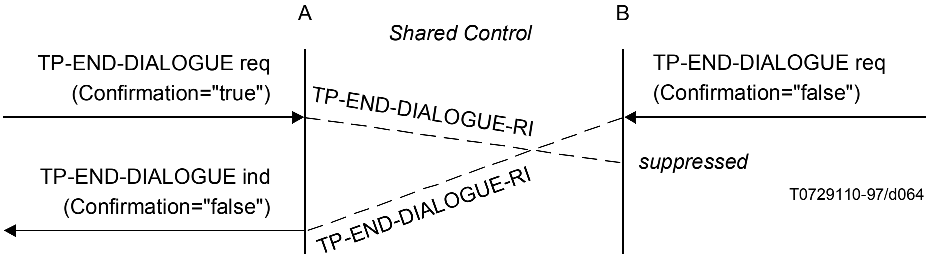


Figure C.59/X.862 – Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-END-DIALOGUE (Confirmation = "false")

C.5.4.2 Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-END-DIALOGUE (Confirmation = "true")

The scenario of Figure C.60 describes a sequence of primitives in the case when a TP-END-DIALOGUE (Confirmation = "true") request collides with another TP-END-DIALOGUE (Confirmation = "true") request in Shared Control.

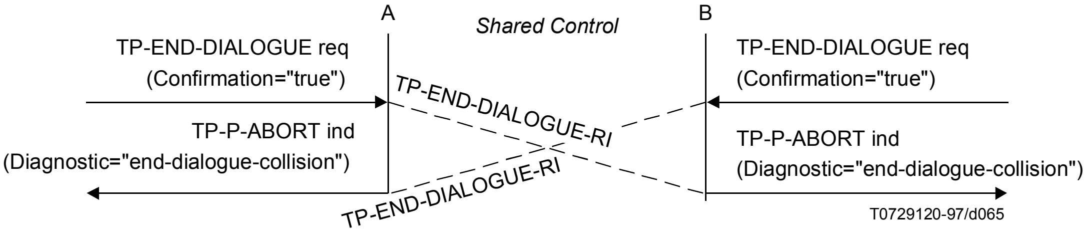


Figure C.60/X.862 – Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-END-DIALOGUE (Confirmation = "true")

C.5.4.3 Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-U-ERROR

The scenario of Figure C.61 describes a sequence of primitives in the case when a TP-END-DIALOGUE (Confirmation = "true") request collides with a TP-U-ERROR request in Polarized Control.

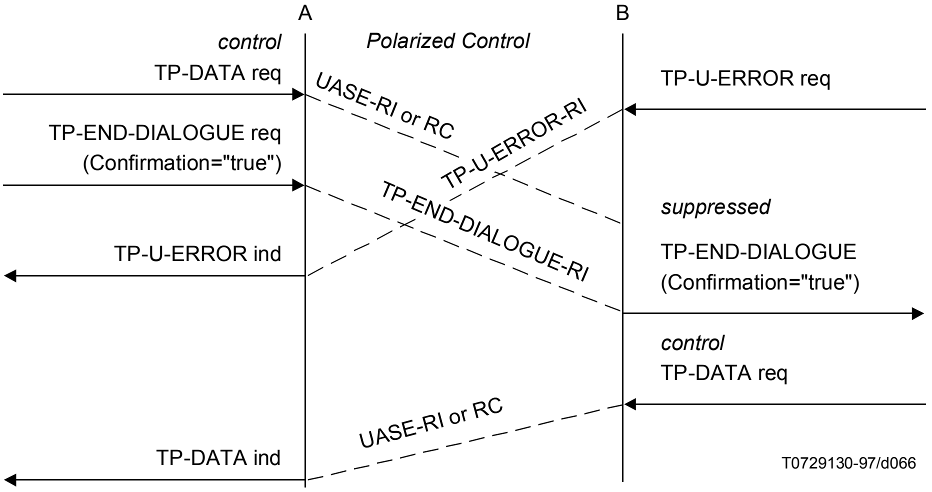


Figure C.61/X.862 – Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-U-ERROR in Polarized Control

The scenario of Figure C.62 describes a sequence of primitives in the case when a TP-END-DIALOGUE (Confirmation = "true") request is involved in a collision with a TP-U-ERROR request in Shared Control. The TP-END-DIALOGUE indication is suppressed.

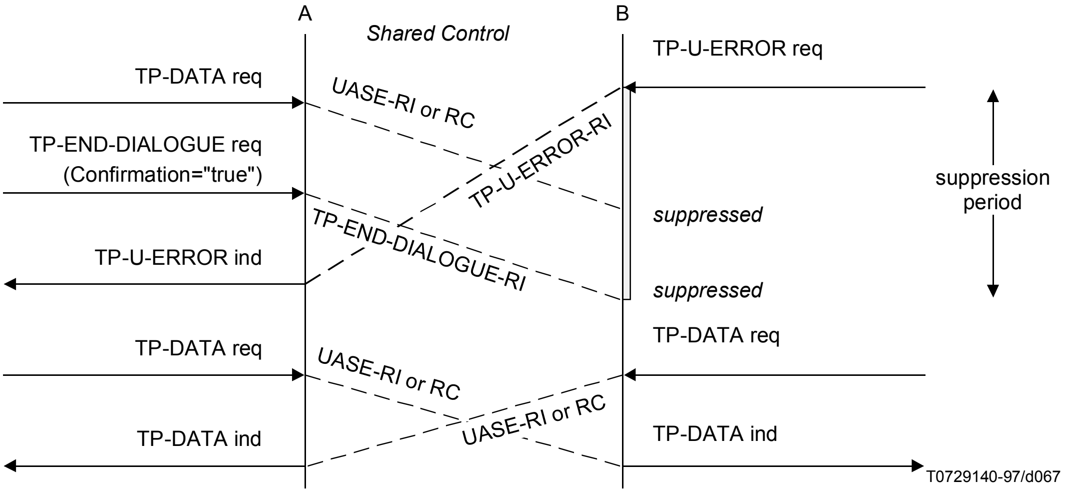


Figure C.62/X.862 – Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-U-ERROR in Shared Control

C.5.4.4 Collision of a TP-END-DIALOGUE (Confirmation = "true") with a TP-REQUEST-CONTROL

The scenario of Figure C.63 describes a sequence of primitives in the case when a TP-END-DIALOGUE (Confirmation = "true") request is involved in a collision with a TP-REQUEST-CONTROL request in Polarized Control.

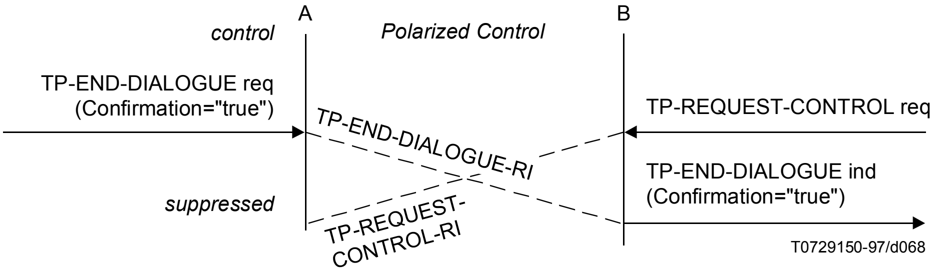


Figure C.63/X.862 – Collision of a TP-END-DIALOGUE (Confirmation = "true") with a TP-REQUEST-CONTROL

C.5.4.5 Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-BEGIN-TRANSACTION

The scenario of Figure C.64 describes a sequence of primitives in the case when a TP-END-DIALOGUE (Confirmation = "true") request is involved in a collision with TP-BEGIN-TRANSACTION in Shared Control.

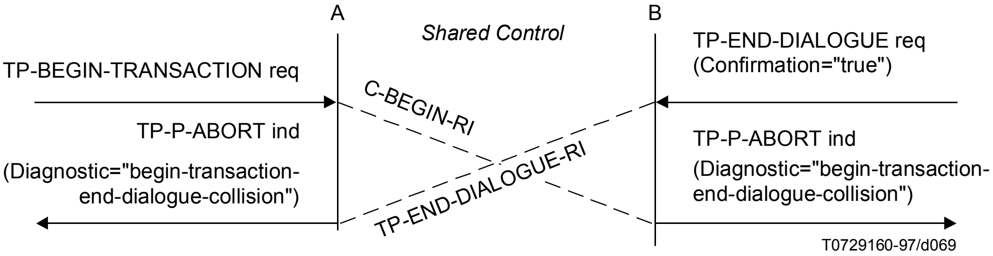


Figure C.64/X.862 – Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-BEGIN-TRANSACTION

C.5.4.6 Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-HANDSHAKE

The scenario of Figure C.65 describes a sequence of primitives in the case when a TP-END-DIALOGUE (Confirmation = "true") request is involved in a collision with TP-HANDSHAKE in Shared Control. TPSUI B shall wait for the TP-HANDSHAKE confirm before issuing the TP-END-DIALOGUE response.

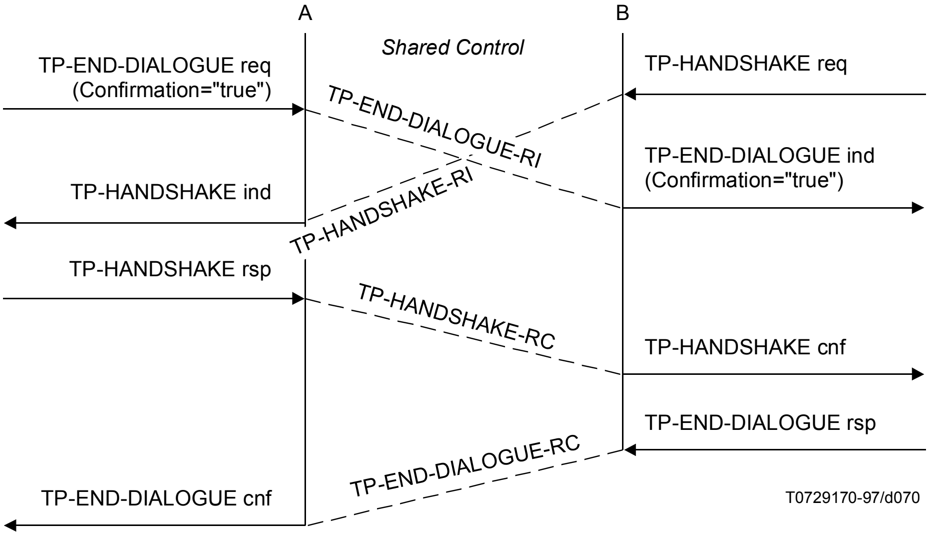


Figure C.65/X.862 – Collision of a TP-END-DIALOGUE (Confirmation = "true") and a TP-HANDSHAKE

C.5.5 Other collisions

C.5.5.1 Collision of a TP-COMMIT and a dialogue establishment reject

The scenarios of Figures C.66 and C.67 show that if a TP-COMMIT request is issued before a TP-BEGIN-DIALOGUE confirm (Result = "rejected"), the transaction is rolled back.

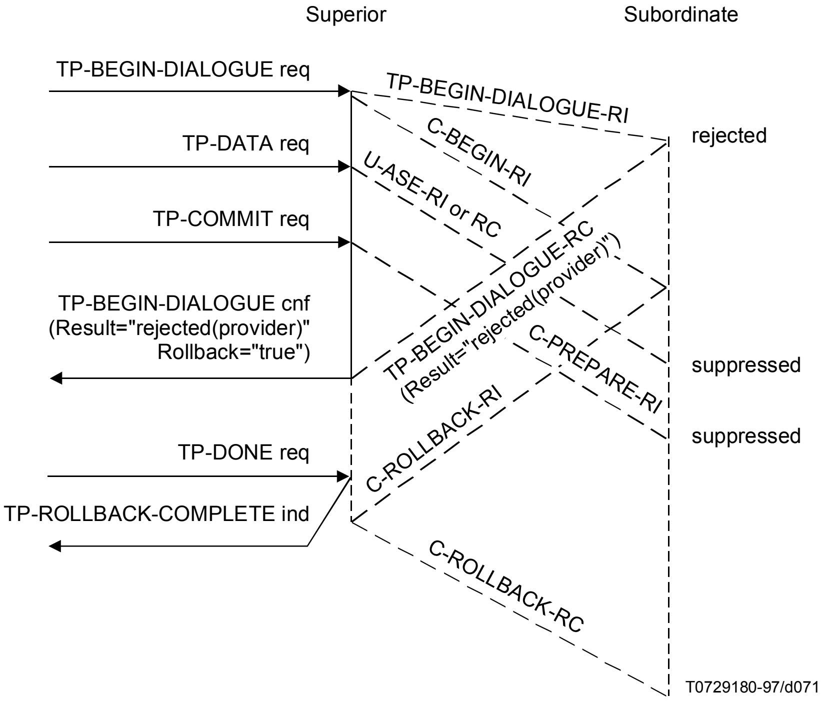
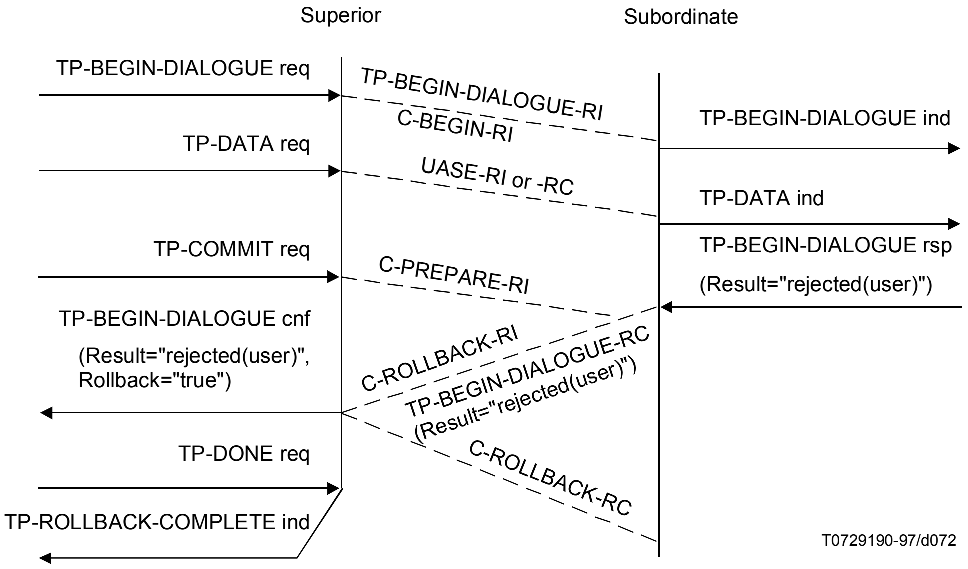


Figure C.66/X.862 – Collision of a TP-COMMIT request and a TP-BEGIN-DIALOGUE confirm (rejected(provider))



NOTE – The C-PREPARE-RI could result in a TP-PREPARE indication at the subordinate before the TP-BEGIN-DIALOGUE response; the final result would be exactly the same.

Figure C.67/X.862 – Collision of a TP-COMMIT request and a TP-BEGIN-DIALOGUE confirm (rejected(user))

C.5.5.2 Collision of a TP-COMMIT and TP-DATA

The scenario of Figure C.68 shows a case where a TP-DATA issued after the Data-Permitted parameter has been set to "true" collides with a TP-COMMIT request; the subordinate is not aware of the collision but the TPSP triggers a rollback because the superior is no longer willing to accept indications from its subordinate.

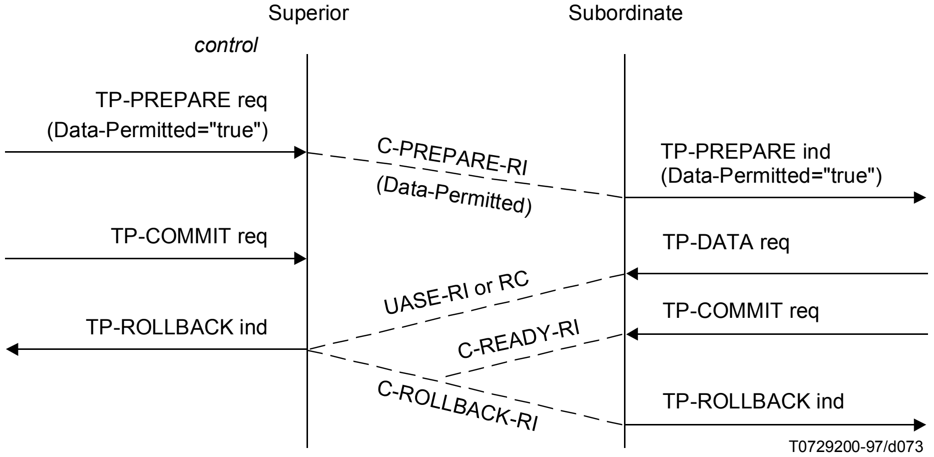


Figure C.68/X.862 – Collision of TP-COMMIT and TP-DATA

C.6 Tree with multiple dialogues (successful cases)

These scenarios show how the commitment procedures operate within a transaction tree containing several intermediate and leaf nodes.

In each case, at the end of the sequences:

– for unchained transactions, the provider-supported transaction has ended and the coordination level is "none";

– for chained transactions, the new transaction will have begun and the transaction tree would retain the same shape as in the previous transaction.

C.6.1 The committing phase driven by TP-COMMIT

The scenario of Figure C.69 shows the sequence of primitives following issuance of the TP-COMMIT request.

C.6.2 The committing phase with TP-PREPARE and TP-READY

The scenario of Figure C.70 shows the sequence of primitives following invocation of the TP-PREPARE request by the root TPSUI for one of its subordinates (C); this illustrates how a TPSUI(TPSUI A) can initiate commitment on some branches (in this case, the single branch A-C) while remaining active, by using the TP-PREPARE request.

The scenario additionally shows the same function being used by node C on its branch C-D.

C.7 Tree with multiple dialogues (unsuccessful cases)

C.7.1 Rollback from the root during the active phase

The scenario of Figure C.71 shows the sequence of primitives following invocation of TP-ROLLBACK request during the active phase of a transaction, for example as the result of a TPSUI error. This scenario covers the case where TP-ROLLBACK request is invoked in the root node of the transaction tree.

At the end of this scenario, all bound data will have been returned to their initial state. If the Chained Transactions functional unit has been selected on the dialogues, a new transaction will have begun.

C.7.2 Rollback from a subordinate during the active phase

The scenarios of Figures C.72 and C.73 are similar to the scenario of Figure C.71; however the rollback originates from an intermediate node. Figures C.72 and C.73 show the unchained and chained cases respectively.

C.7.3 Dialogue abort during phase I of commitment

The scenario of Figure C.74 is based on scenario of Figure C.69 except that a communications failure (causing TP-P-ABORT indications) occurs between A and B, and C and D, during the first phase of commitment.

As the failure between C and D occurs before the commitment sequence begins, this is equivalent to a dialogue abort in the active phase and so no later recovery actions are required. Between B and A, transaction recovery is initiated from B and takes place on a new association.

At the end of the sequence, the dialogue tree contains only A and C; then, depending on the functional units selected:

– Unchained Transactions: The transaction has been rolled back and no new transaction has been started.

– Chained Transactions: The transaction has been rolled back and a new transaction tree established containing A and C.

Further action is determined by the TPSUIs themselves.

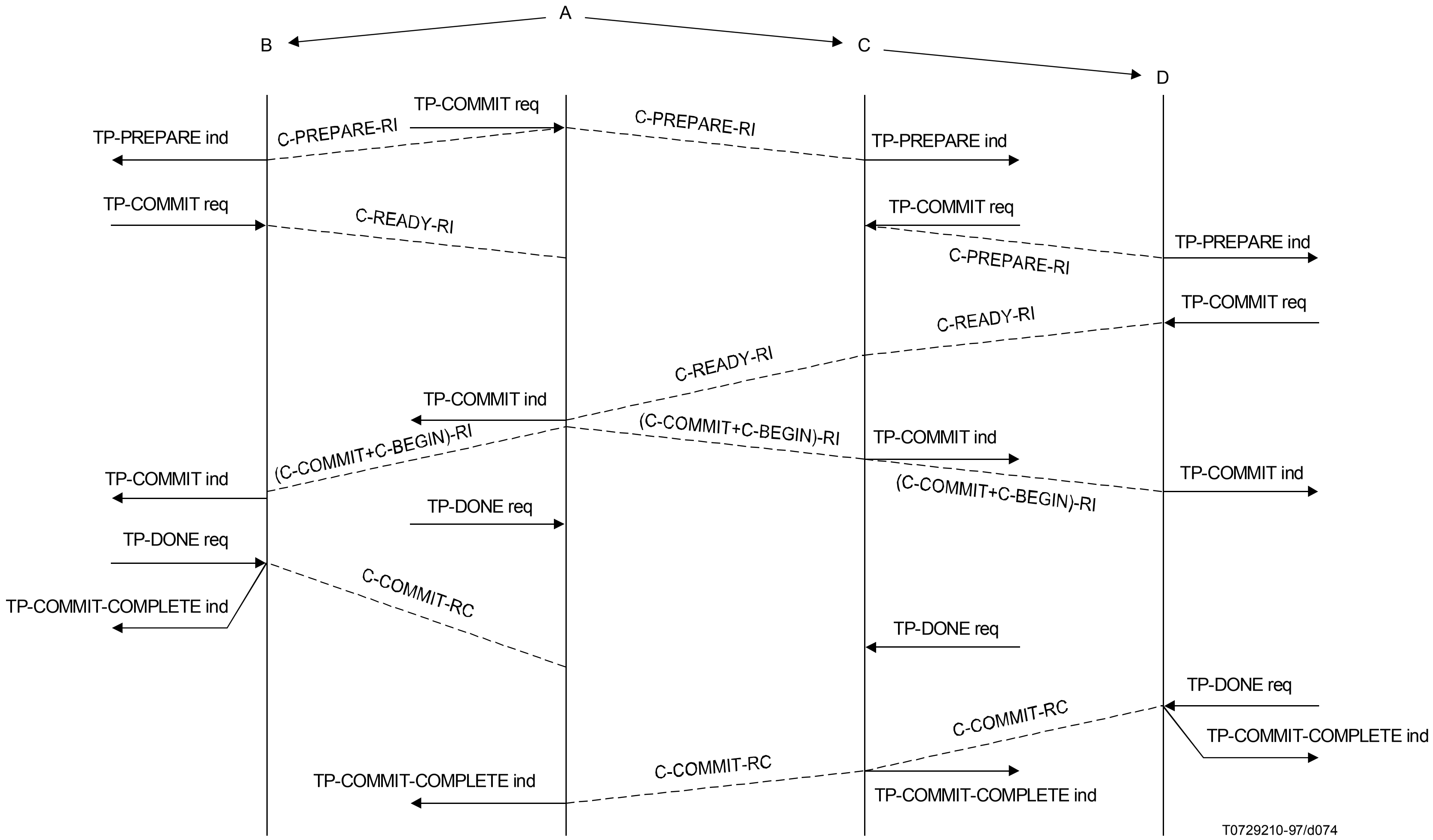


Figure C.69/X.862 – Commitment in a multi-dialogue tree (Chained Transactions)

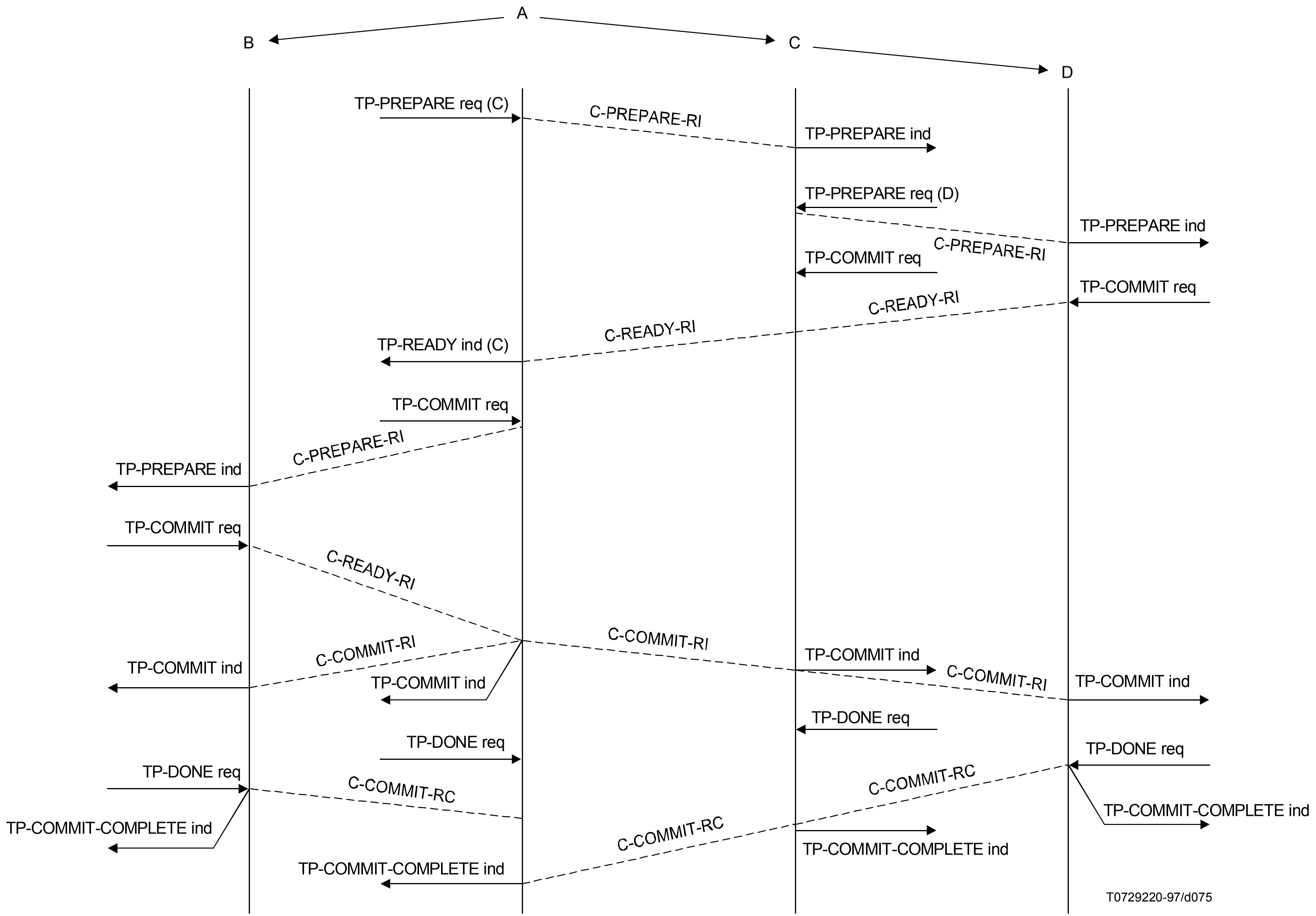


Figure C.70/X.862 – Commitment in a multi-dialogue tree with use of TP-PREPARE (Unchained Transactions)

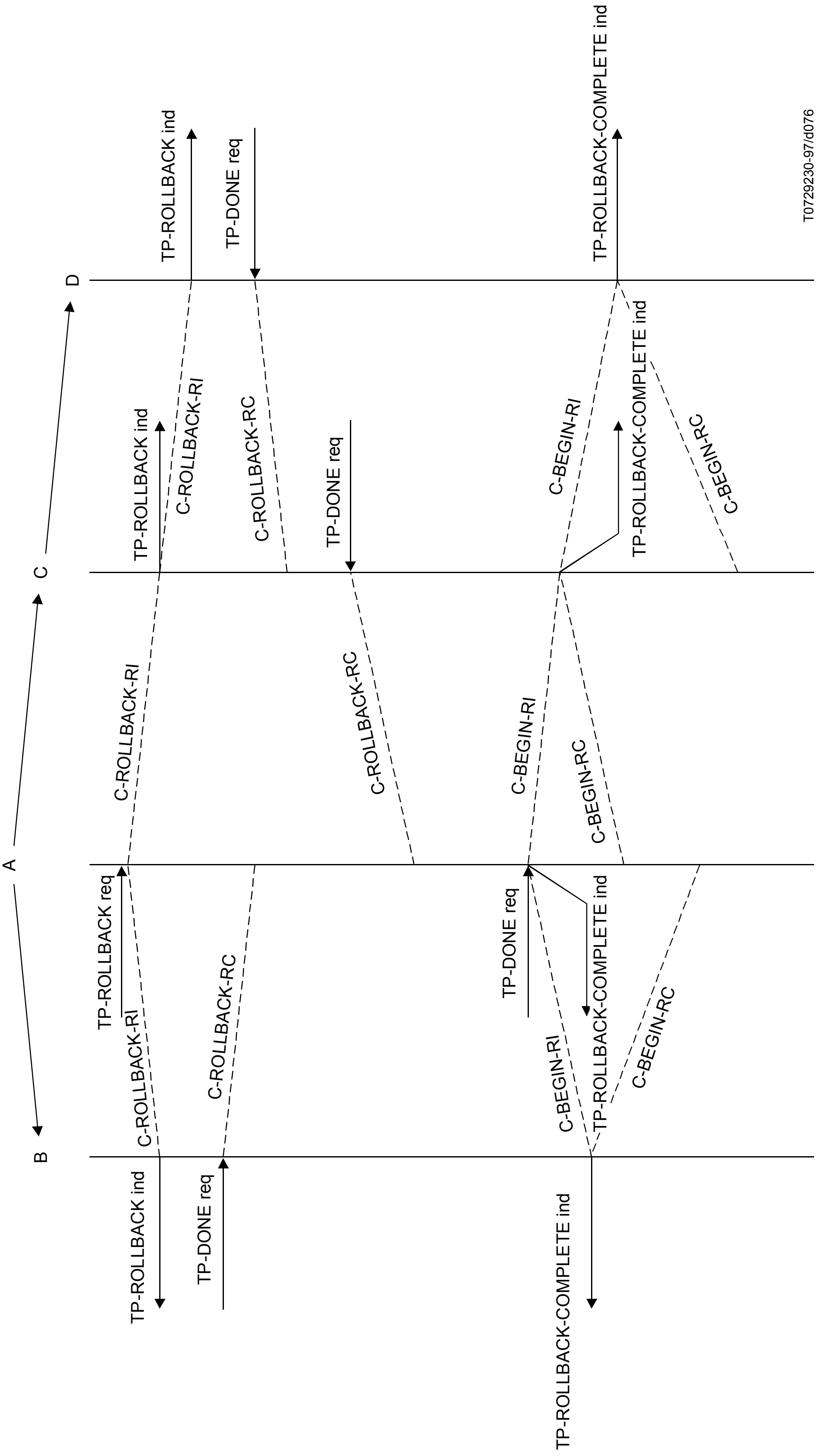


Figure C.71/X.862 – Rollback from a root node (Chained Transactions)

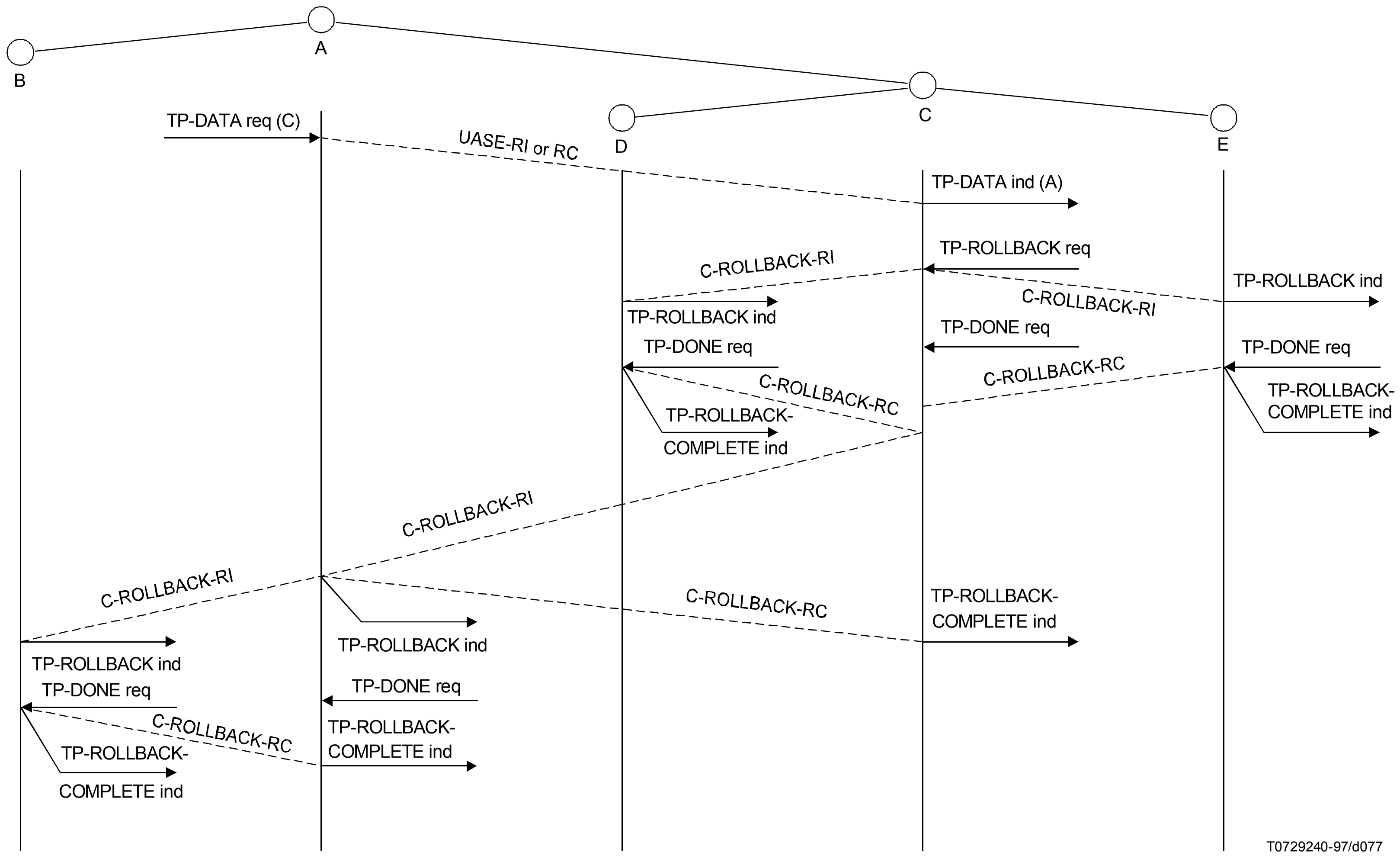


Figure C.72/X.862 – Rollback from an intermediate node (Unchained Transactions)

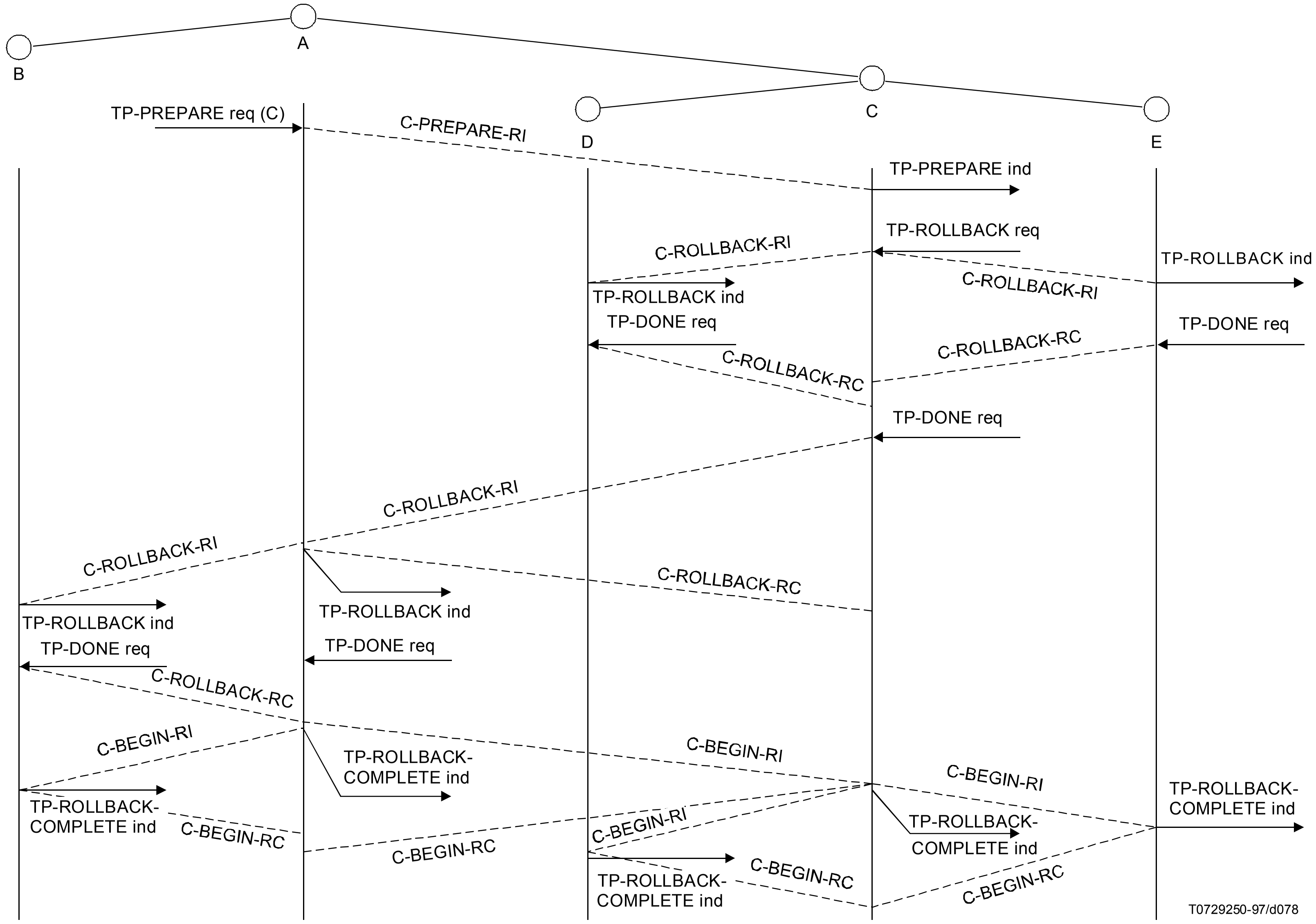


Figure C.73/X.862 – Rollback from an intermediate node (Chained Transactions)

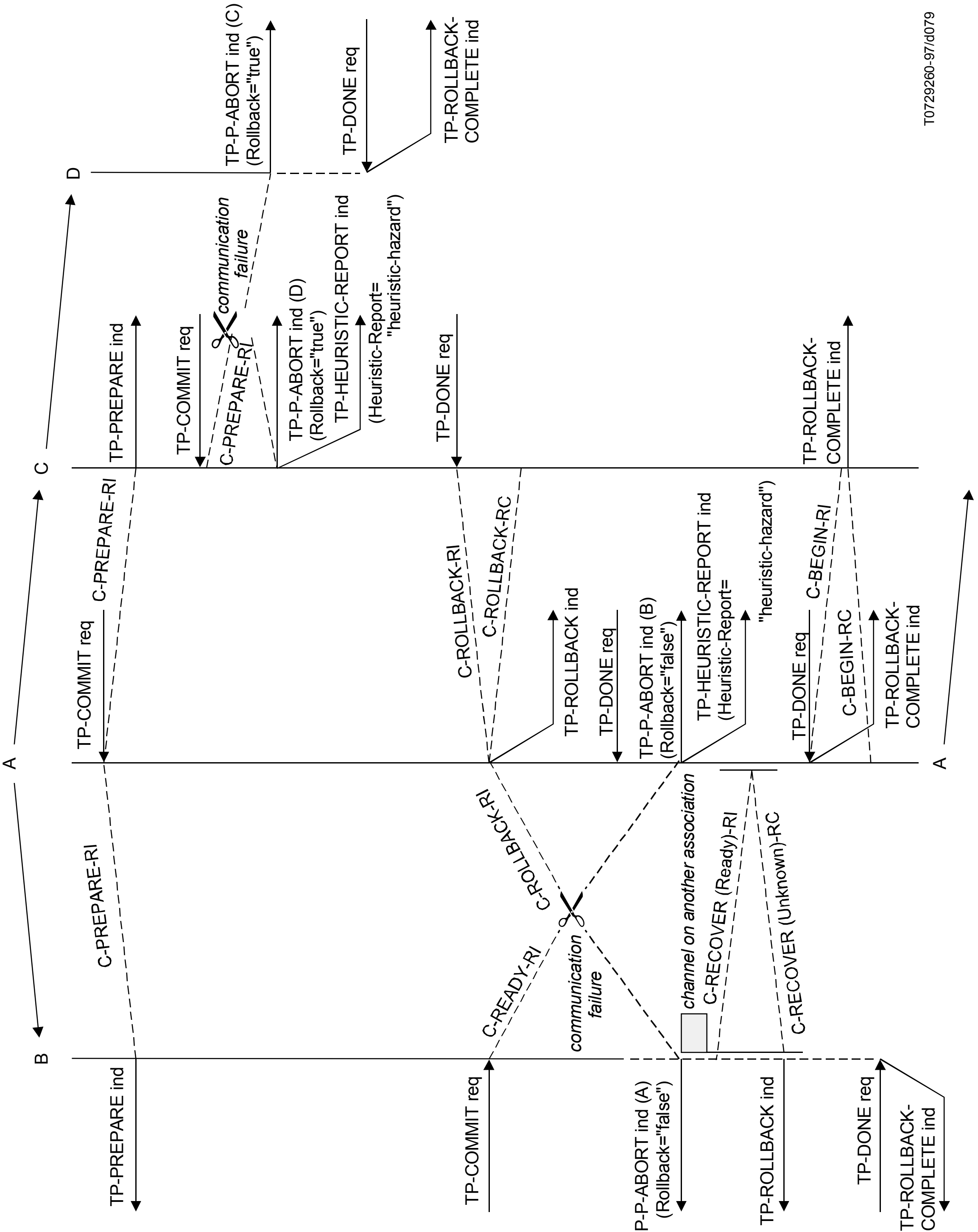


Figure C.74/X.862 – Two failures during phase I of commitment (Chained Transactions)

C.7.4 Rollback-related actions

The scenario of Figure C.75 shows a rollback initiated from a leaf node. In reaction to the rollback, two other leaf nodes abort their superior dialogue.

In turn, their superior does not accept the situation and aborts its superior dialogue.

At the end of the scenario, the tree is limited to two nodes.

C.7.5 Dialogue abort during the active phase

In the scenario of Figure C.76, the dialogue between C and E is aborted during the active phase of the provider-supported transaction in which A, B, C, D, E and E's subordinates are participating. In this case, this provider-supported transaction is rolled back (TP-P-ABORT indication is issued with the Rollback parameter set to "true" to both C and E).

At the end of the scenario, two isolated transaction trees exist:

1) a transaction tree with A as the root and which consists of A, B, C and D;

2) a transaction tree with E as the root.

C.7.6 Dialogue abort during the active phase – Transaction tree above the failure disbanded

The beginning of the scenario of Figure C.77 is similar to the beginning of the previous one. But after being aware of the failure with its superior, C aborts the dialogues it has with its superior (A) and its subordinate (D) by issuing TP-U-ABORT request on these dialogues. A, in return, aborts the dialogue with its last subordinate (B).

At the end of this scenario, the transaction tree superior to the failure is completely disbanded and A, B, C and D are isolated. E has been isolated from C by the failure but is still the root of a transaction tree.

C.7.7 Dialogue abort during the second phase of commitment

In the scenario of Figure C.78, the failure between C and E occurs after C has received TP-COMMIT indication, but before E has received TP-COMMIT indication. Therefore, the TP-P-ABORT indication issued to C and E is delivered with the Rollback parameter set to "false" (C knows that the terminating provider-supported transaction is committed, E is in doubt).

After recovery takes place, E receives TP-COMMIT indication and completes the commitment of its subtree.

Since the dialogue between C and E was in chained mode, the next transaction has to be rolled back from node C.

At the end of the scenario, two isolated Transaction trees exist:

1) a transaction tree with A as the root and which consists of A, B, C and D;

2) a transaction tree with E as the root.

C.7.8 Dialogue abort during the second phase of commitment – Subordinate of the failed dialogue aborts its other dialogues

In the scenario of Figure C.79, the failure between A and C occurs after A has received TP-COMMIT indication, but before C has received TP-COMMIT indication. Therefore, the TP-P-ABORT indication issued to A and C is delivered with the Rollback parameter set to "false" (A knows that the terminating provider-supported transaction is committed, C is in doubt).

After having received TP-P-ABORT indication, C decides to abort the dialogue it has with its two subordinates (D and E) by issuing TP-U-ABORT request on this dialogue.

Since the dialogue between A and C was in chained mode, the next transaction has to be rolled back from node A.

At the end of the scenario, C, D and E are isolated, A and B form a transaction tree with A as the root.

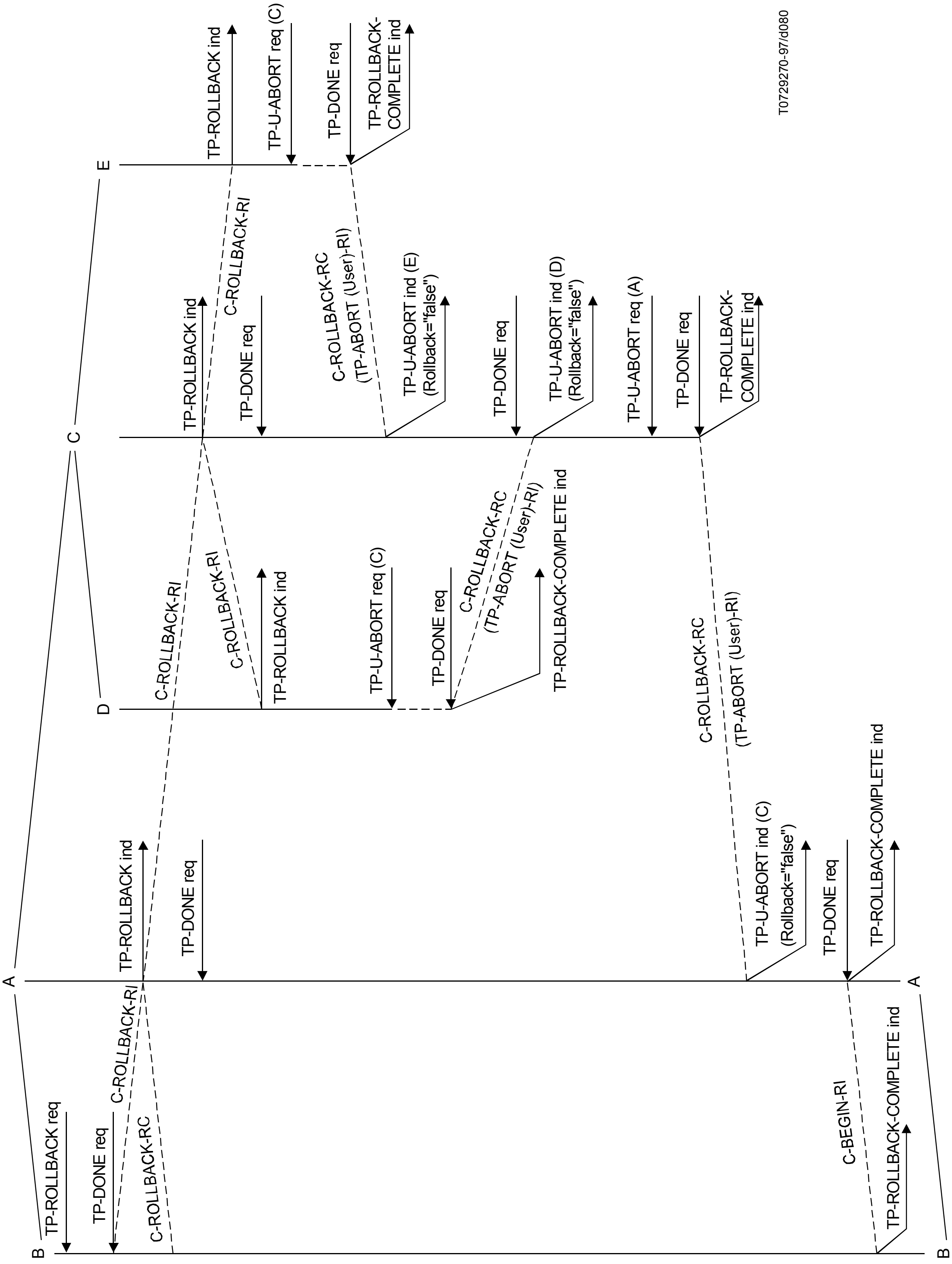


Figure C.75/X.862 – Rollback-related actions – A and B nodes remain (Chained Transactions)

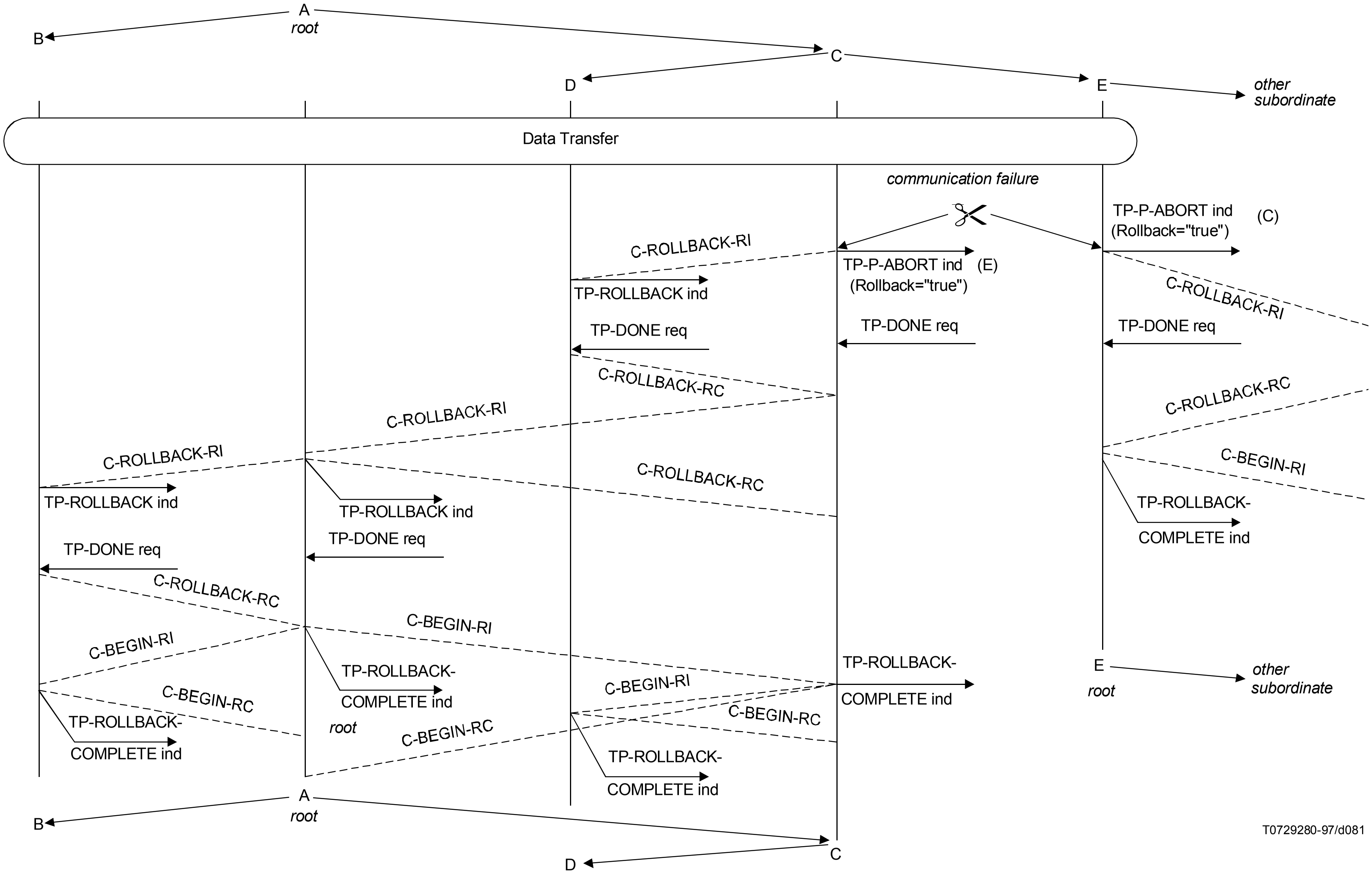


Figure C.76/X.862 – Failure during phase I – Two isolated transaction trees (Chained Transactions)

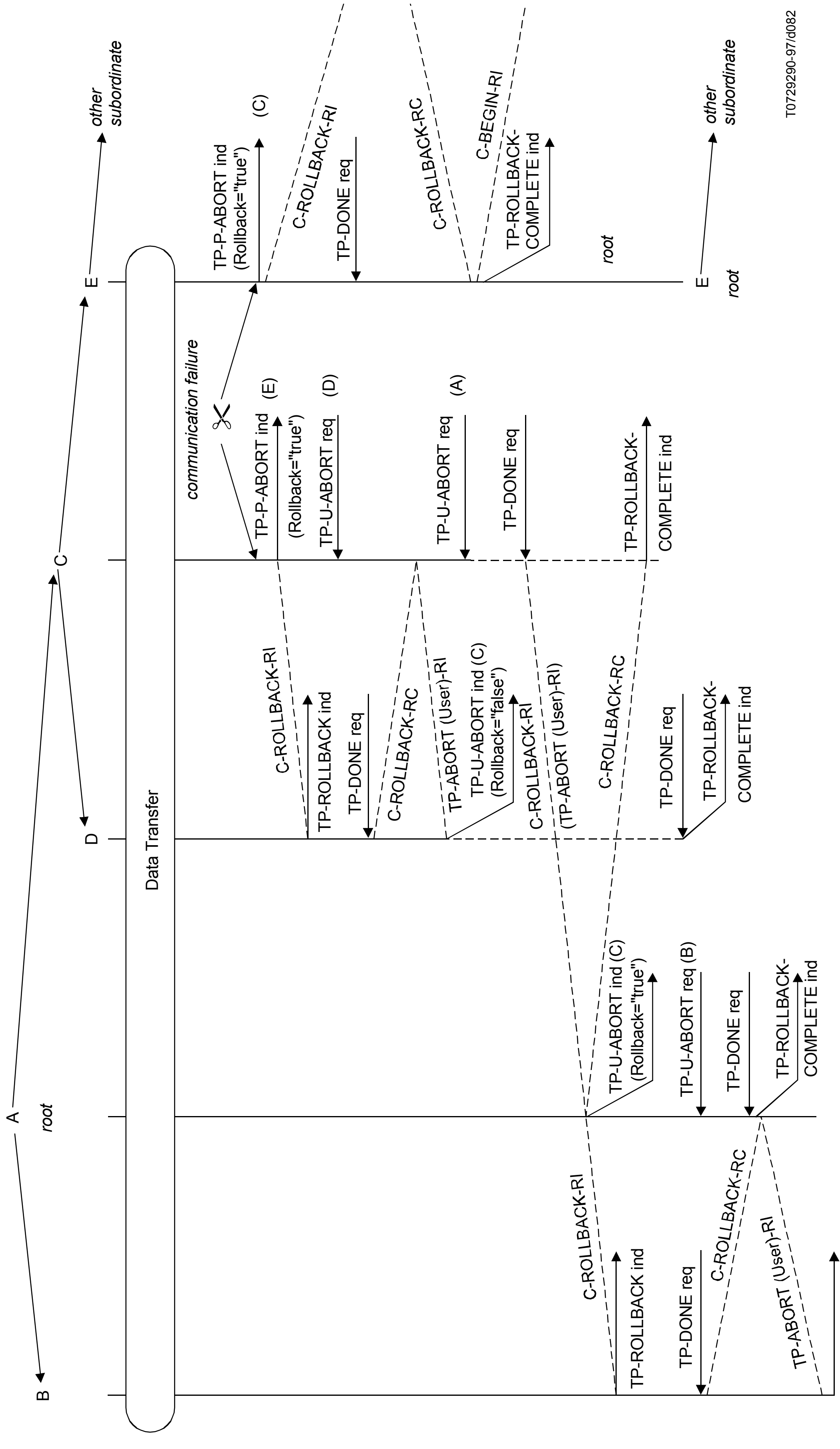


Figure C.77/X.862 – Failure during phase I – Superior tree aborted (Chained Transactions)

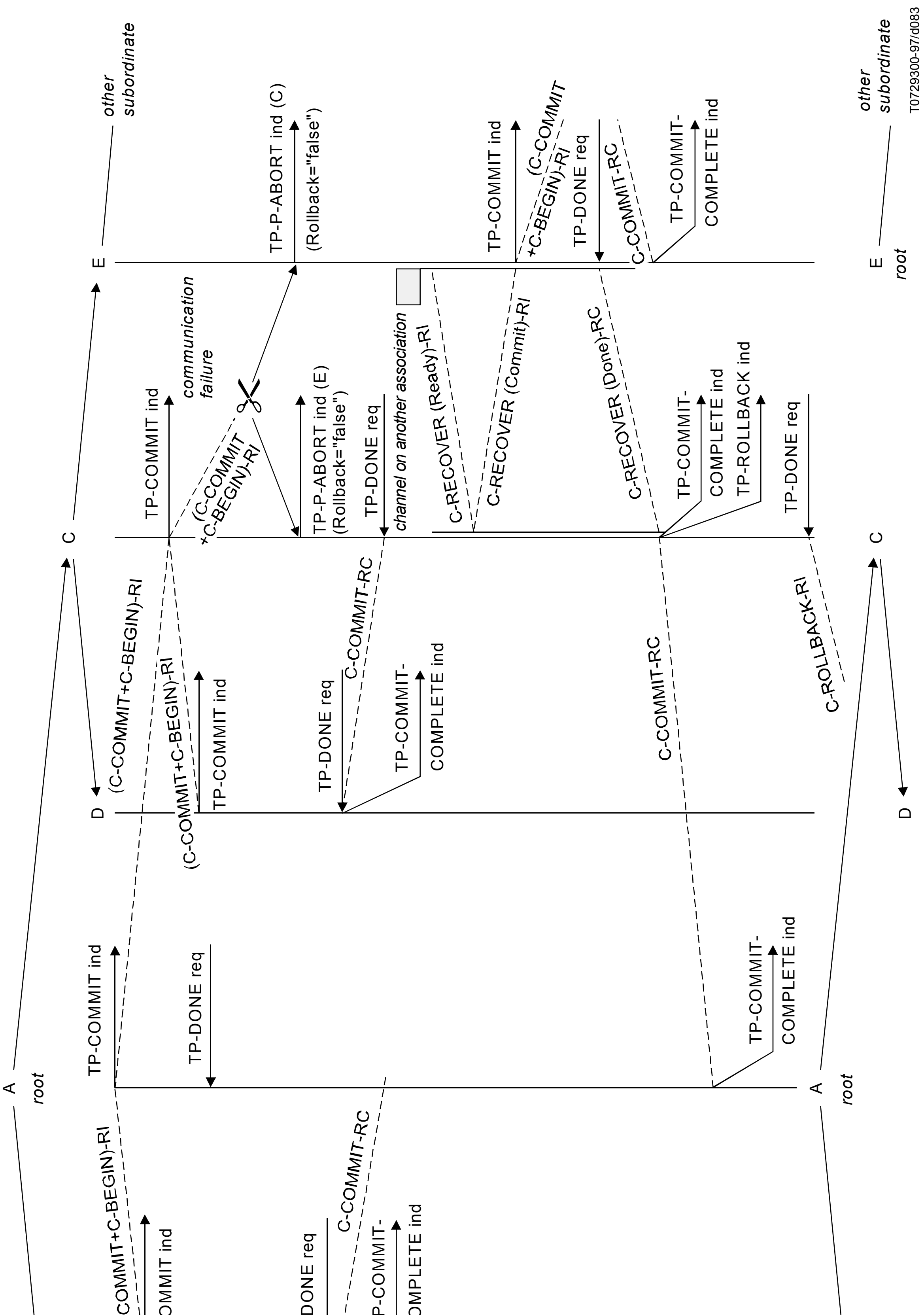


Figure C.78/X.862 – Failure during phase II – Two isolated transaction trees (Chained Transactions)

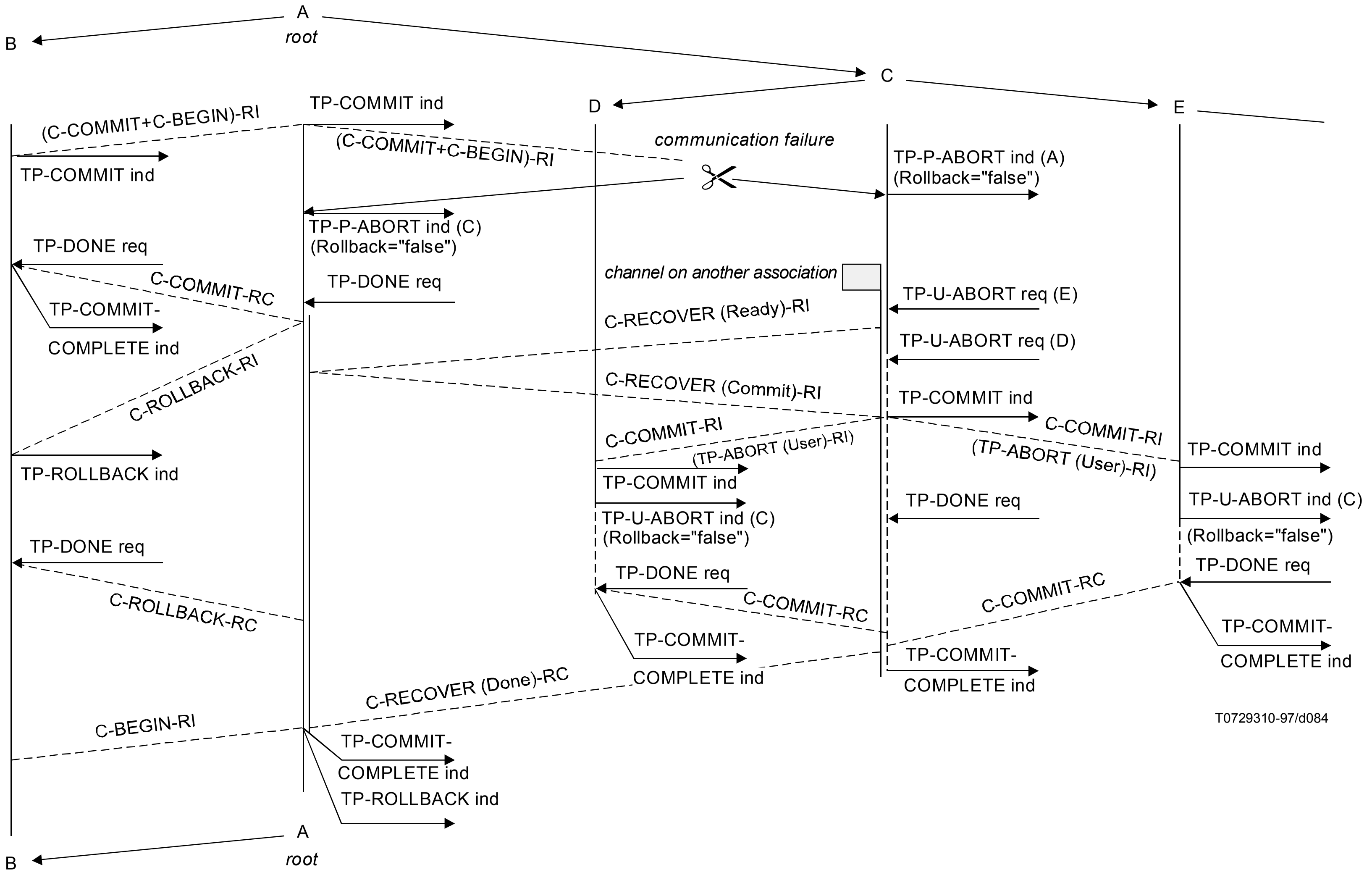


Figure C.79/X.862 – Failure during phase II – Subordinate refused to become root (Chained Transactions)

C.7.9 Dialogue abort with the superior after TP-COMMIT indication has been issued

In the scenario of Figure C.80, a failure occurs between A and its subordinate C, after C has received TP-COMMIT indication; in this case, the next provider-supported transaction has to be rolled back in the subtree which is isolated from the root of the transaction tree (C, D and E).

D and E, which have already received TP-COMMIT-COMPLETE indication and may have begun to perform some actions in the context of the next provider-supported transaction, receive TP-ROLLBACK indication and enter the rollback procedure. A enters the rollback procedure as well.

At the end of the scenario, C is the root of a transaction tree consisting of C, D and E. Another transaction tree consisting of A and B also exists.

C.7.10 Dialogue abort with the superior after TP-COMMIT indication has been issued – Node subordinate to the failed dialogue aborts its other dialogues

The beginning of the scenario of Figure C.81 is similar to the beginning of the previous one. But after being aware of the failure with its superior, C decides to abort the dialogue it has with its two subordinates (D and E) by issuing TP-U-ABORT request on this dialogue.

The next transaction is rolled back in both transaction trees.

At the end of the scenario, C, D and E are isolated, A and B form a transaction tree with A as the root.

C.7.11 Dialogue abort with the superior after TP-COMMIT indication has been issued – Node superior to the failed dialogue aborts its other dialogues

In the scenario of Figure C.82, a failure occurs between C and its subordinate E, after C has received TP-COMMIT indication. E has not received TP-COMMIT indication and will receive it after recovery.

D and E, which have already received TP-COMMIT-COMPLETE indication and can have begun to perform some actions in the context of the next provider-supported transaction, receive TP-ROLLBACK indication and enter the rollback procedure. A also enters the rollback procedure.

C decides to abort the dialogues it has with its superior (A) and its subordinate (D) by issuing TP-U-ABORT request on these dialogues.

At the end of the scenario, C and D are isolated, A and B form a transaction tree with A as the root, and the E, since it has not performed any action related to this failure, is the root of a new transaction tree.

C.7.12 Dialogue abort with the superior after TP-COMMIT-COMPLETE indication has been issued – Node superior to the failed dialogue aborts its other dialogues

In the scenario of Figure C.83, a failure occurs between B and its subordinate C, after C has received TP-COMMIT-COMPLETE indication.

C receives the TP-P-ABORT indication with the Rollback parameter set to "true" while being in the next transaction.

B receives the TP-P-ABORT indication with the Rollback parameter set to "false" before receiving TP-COMMIT-COMPLETE indication, but the next transaction is rolled back.

C.8 Heuristic decisions and reporting

In all cases other than scenario of Figure C.90, a subordinate has issued TP-COMMIT request and then has not received either TP-COMMIT indication or TP-ROLLBACK indication. Its patience exhausted, the subordinate takes a heuristic decision. The following scenarios are provided:

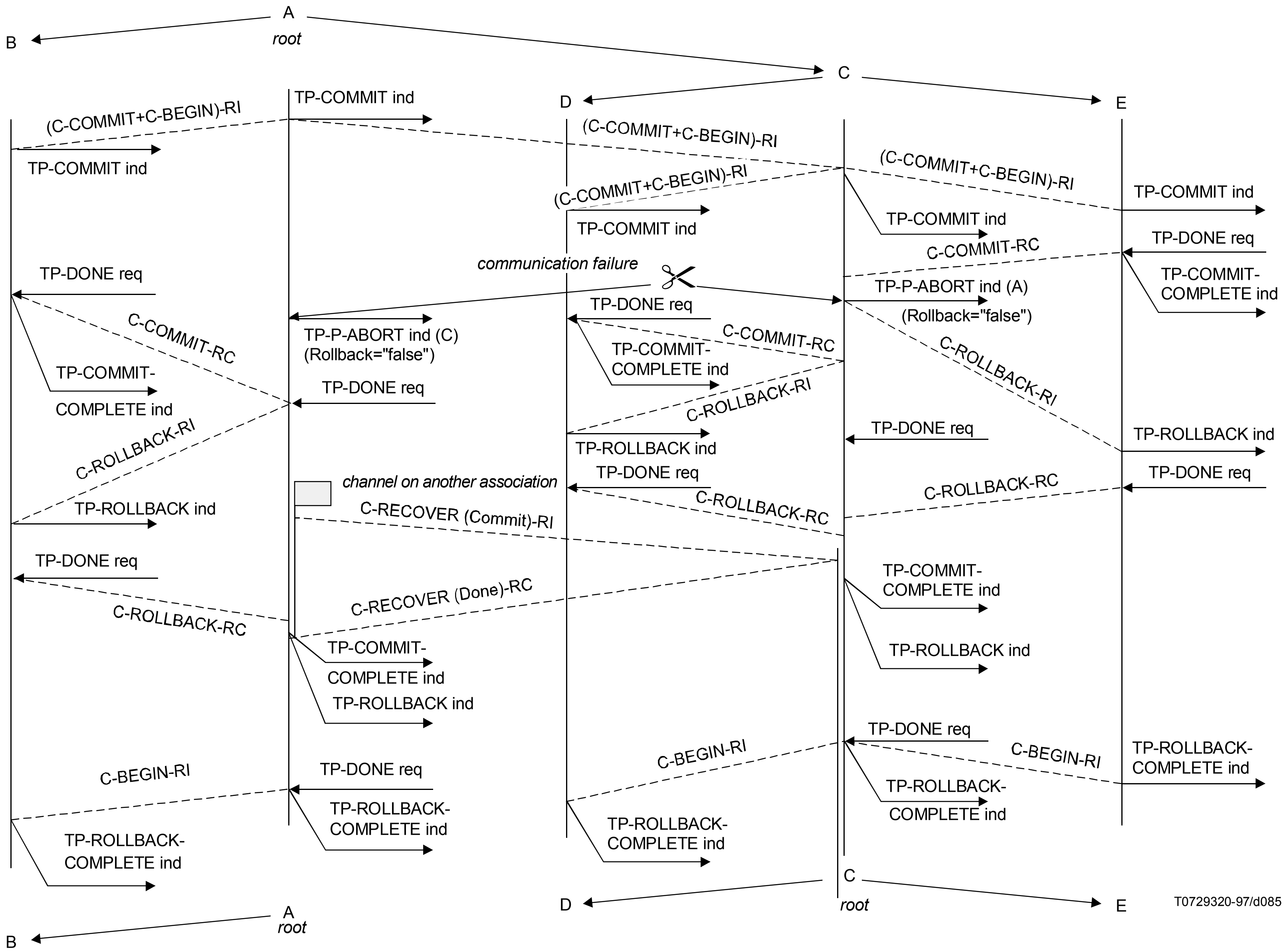


Figure C.80/X.862 – Failure during phase II – Isolated node becomes root (Chained Transactions)

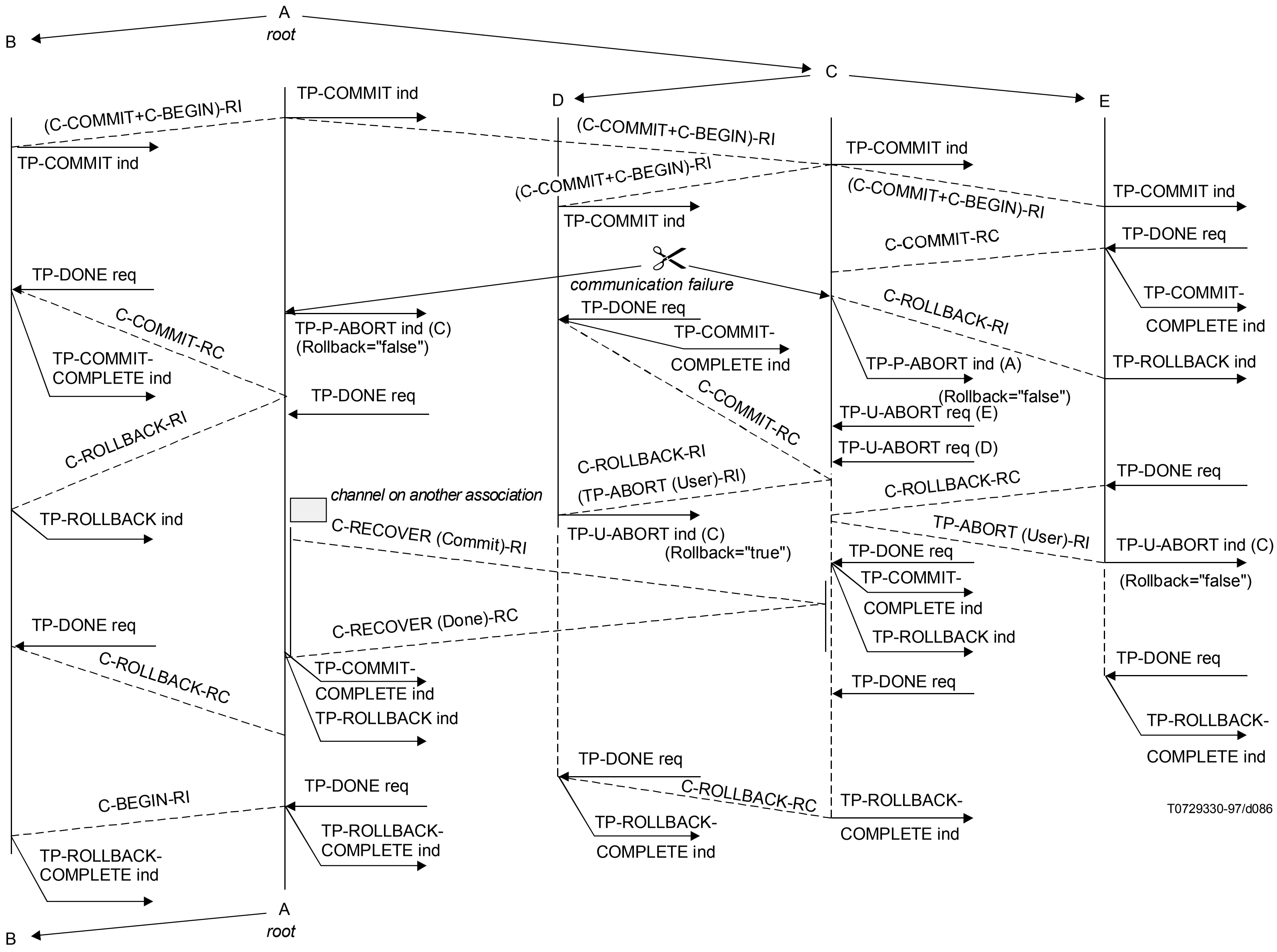


Figure C.81/X.862 – Failure during phase II – Intermediate aborts all dialogues (Chained Transactions)

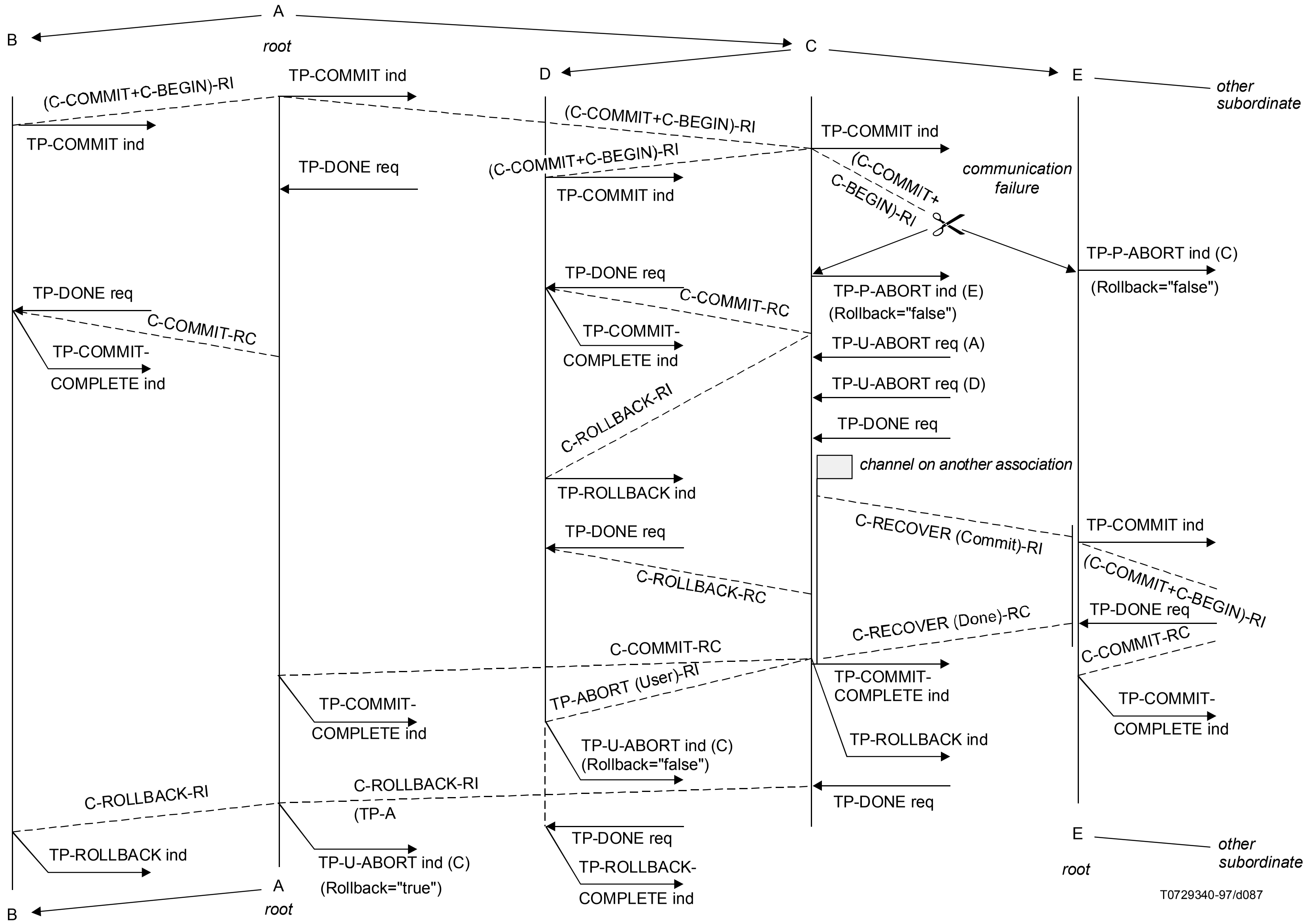


Figure C.82/X.862 – Failure during phase II – Failure after subordinate has issued TP-COMMIT-COMPLETE indication

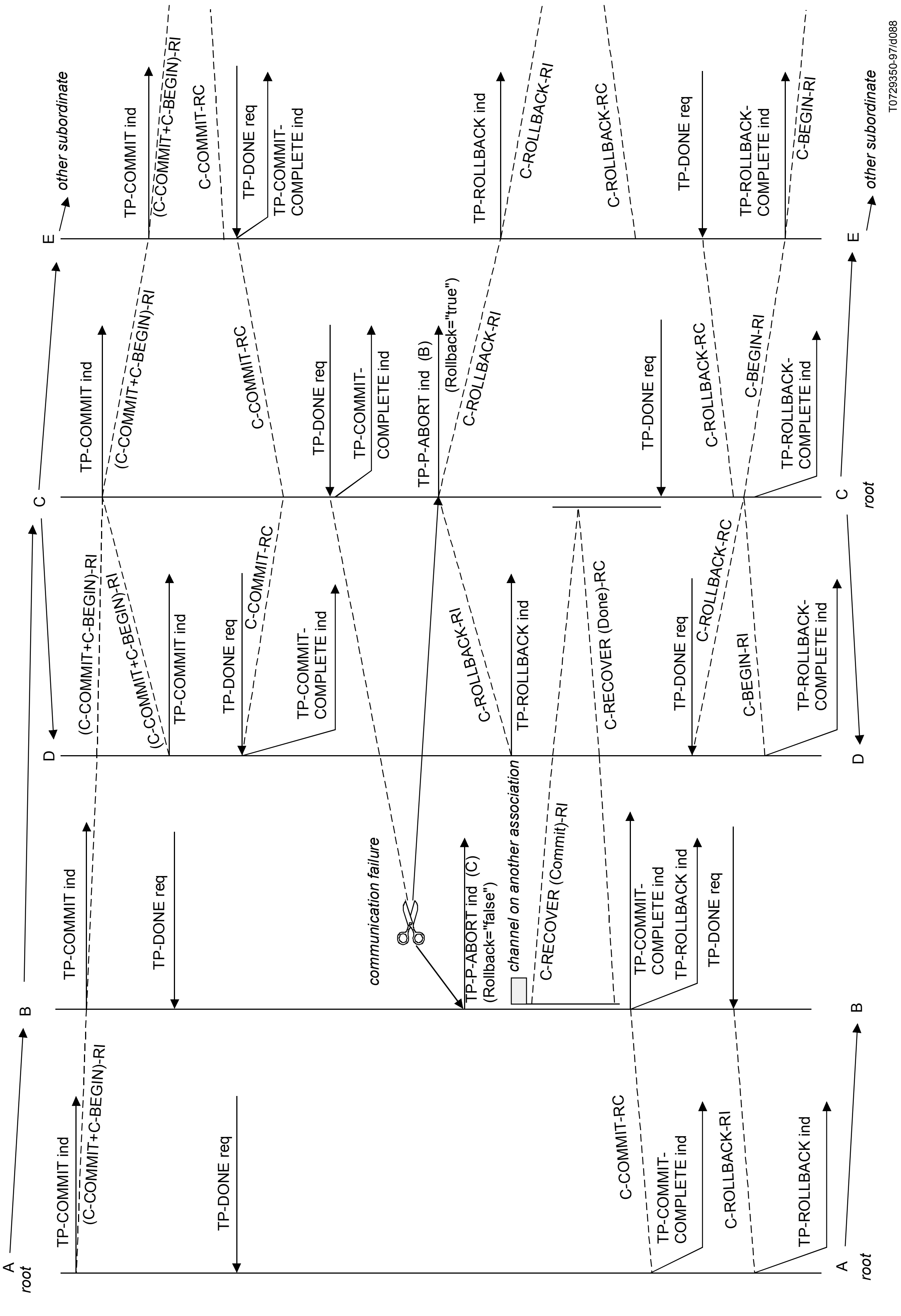


Figure C.83/X.862 – Failure during phase II – Failure after subordinate has issued TP-COMMIT-COMPLETE indication

C.8.1 Heuristic decisions when the outcome of the transaction is commit

C.8.1.1 Subordinate rolls back – Mix detected

The scenario of Figure C.84 shows a case where the subordinate decides to heuristically roll back; since the outcome is commit, a recovery occurs and a heuristic-mix report is given to the superior.

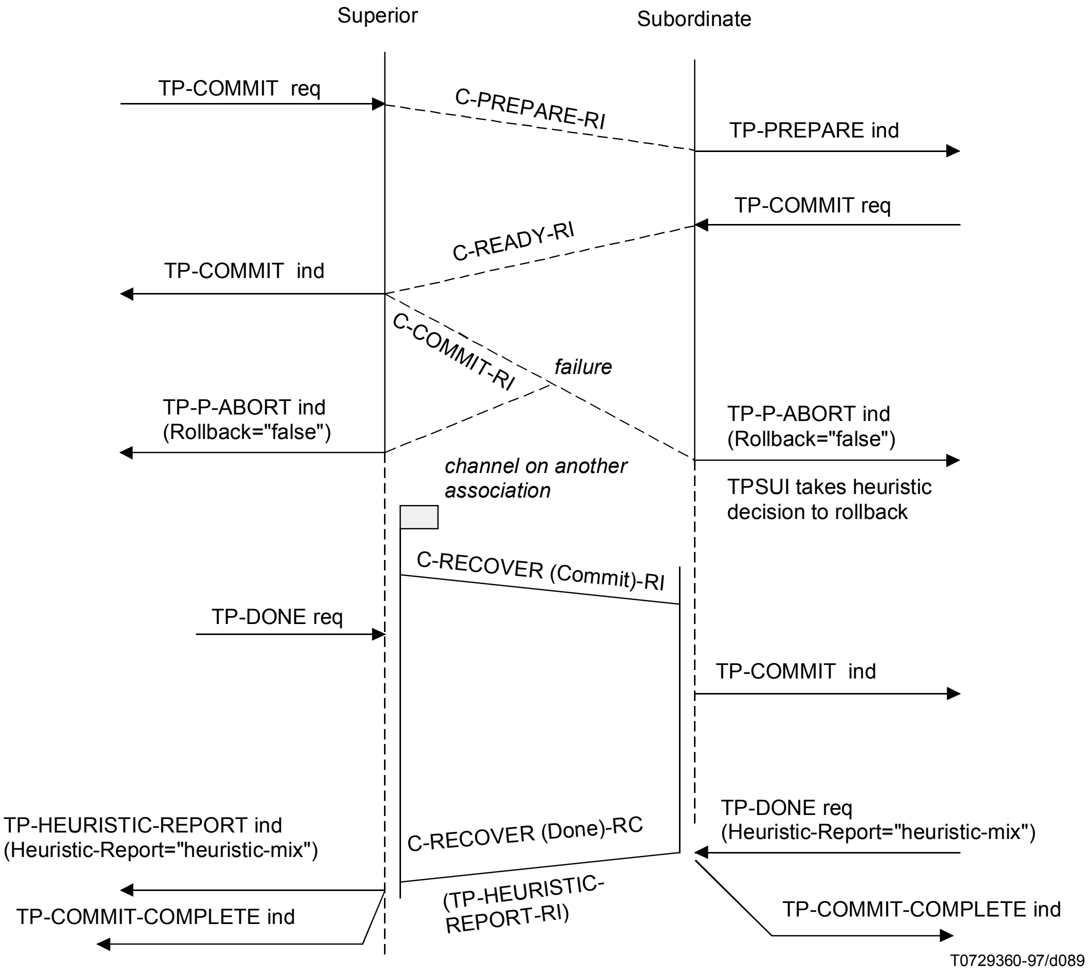


Figure C.84/X.862 – Dialogue aborted during commitment – Subordinate takes (wrong) decision to roll back

C.8.1.2 Subordinate commits – No damage

The scenario of Figure C.85 shows a case where the subordinate decides to heuristically commit; since the outcome is commit, a recovery occurs and no damage exists.

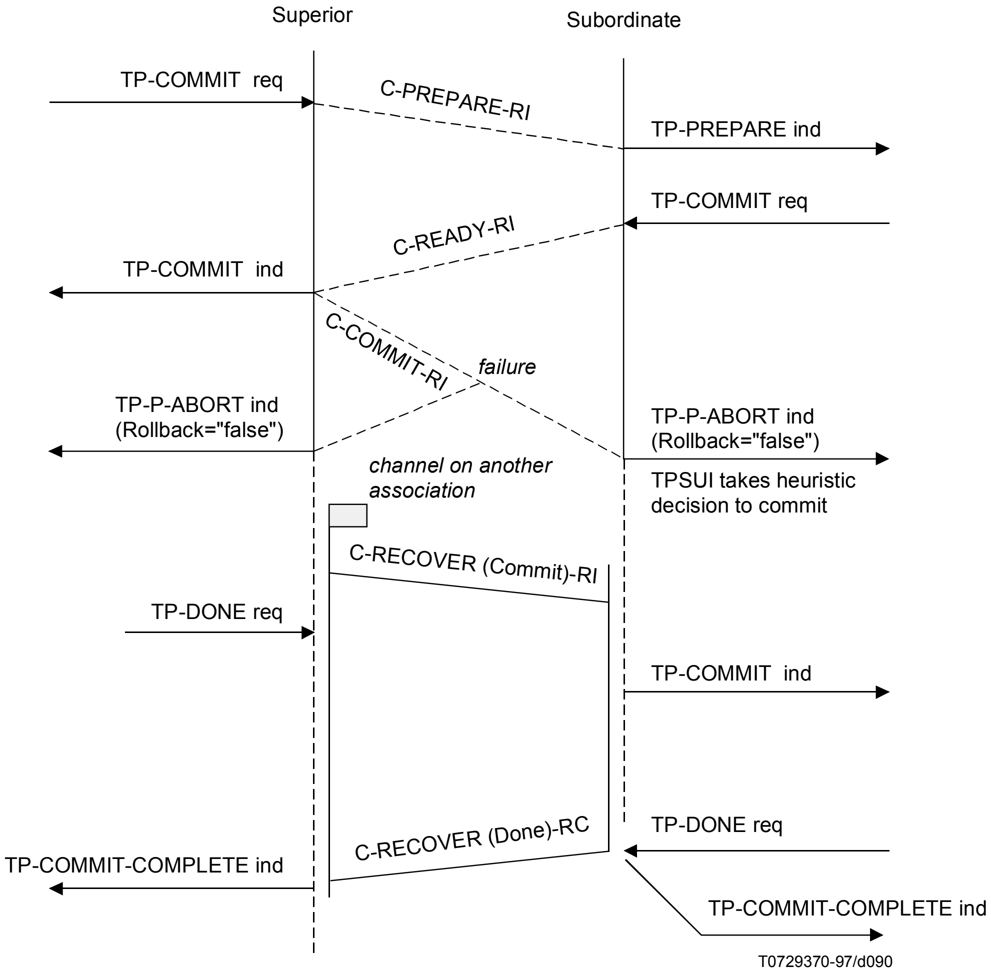


Figure C.85/X.862 – Dialogue aborted during commitment – Subordinate takes (correct) decision to commit

C.8.1.3 Subordinate rolls back but is able to compensate – No damage

The scenario of Figure C.86 shows a case where the subordinate decides to heuristically roll back; since the outcome is commit, a recovery occurs, but, since the subordinate is able to compensate, no damage is reported.

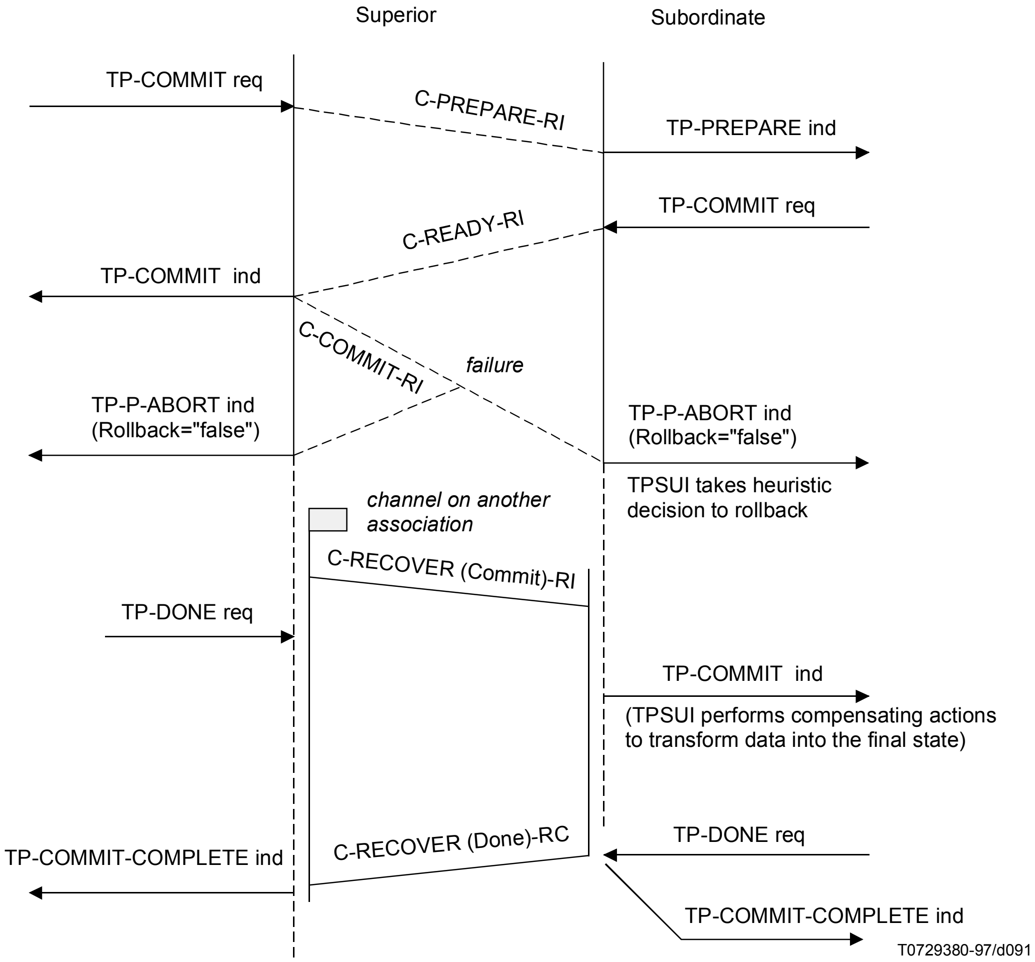


Figure C.86/X.862 – Dialogue aborted during commitment – Subordinate takes (wrong) decision to roll back but is able to compensate

C.8.1.4 Subordinate TPSUI commits, TPSP rolls back – Mix detected

The scenario of Figure C.87 shows a case where the subordinate TPSUI decides to heuristically commit whereas the TPSP decides to roll back; since the outcome is commit, a recovery occurs and a heuristic-mix report is given to the superior.

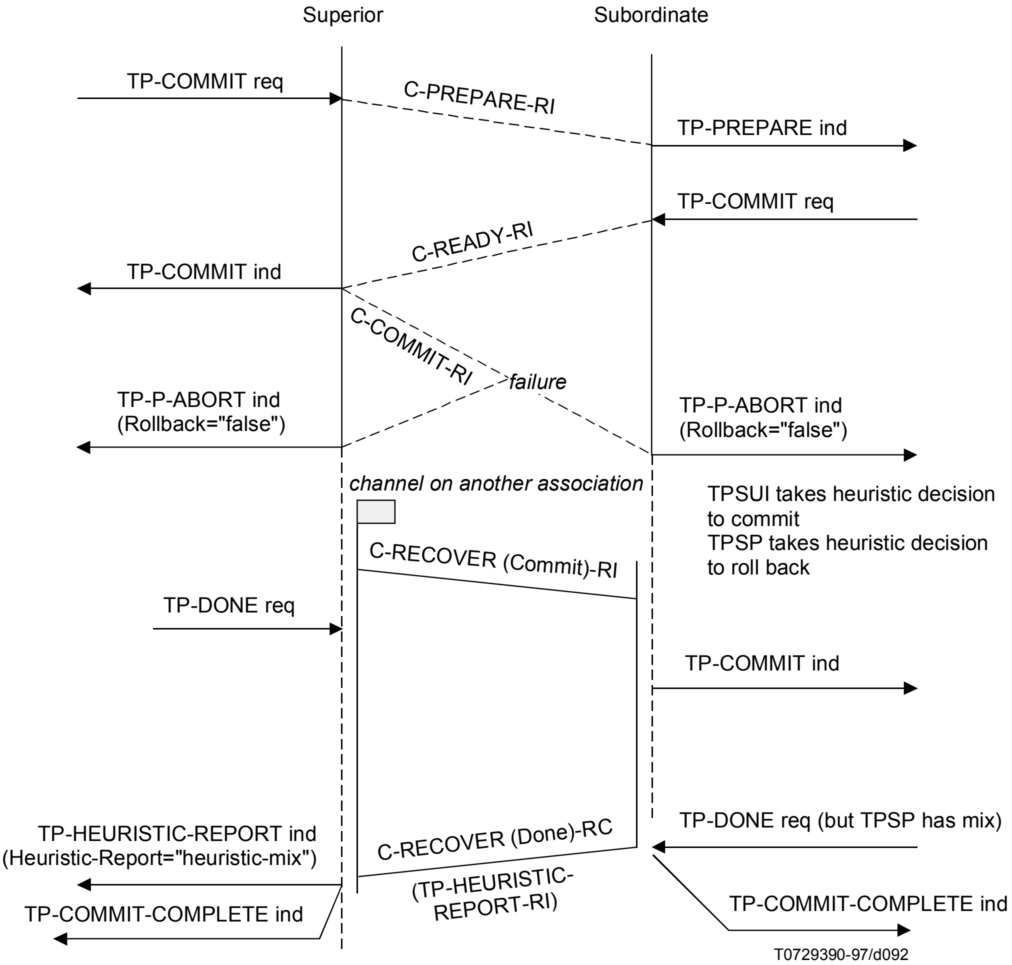


Figure C.87/X.862 – Dialogue aborted during commitment – Subordinate takes contradicting heuristic decisions producing an internal mix

C.8.2 Heuristic decisions when the outcome of the transaction is rollback

C.8.2.1 Subordinate commits – Heuristic-hazard reported

The scenario of Figure C.88 shows a case where the subordinate decides to heuristically commit; since the dialogue has failed with the superior before TP-COMMIT indication is issued, the superior rolls back and no recovery occurs, a heuristic-hazard report is given to the superior.

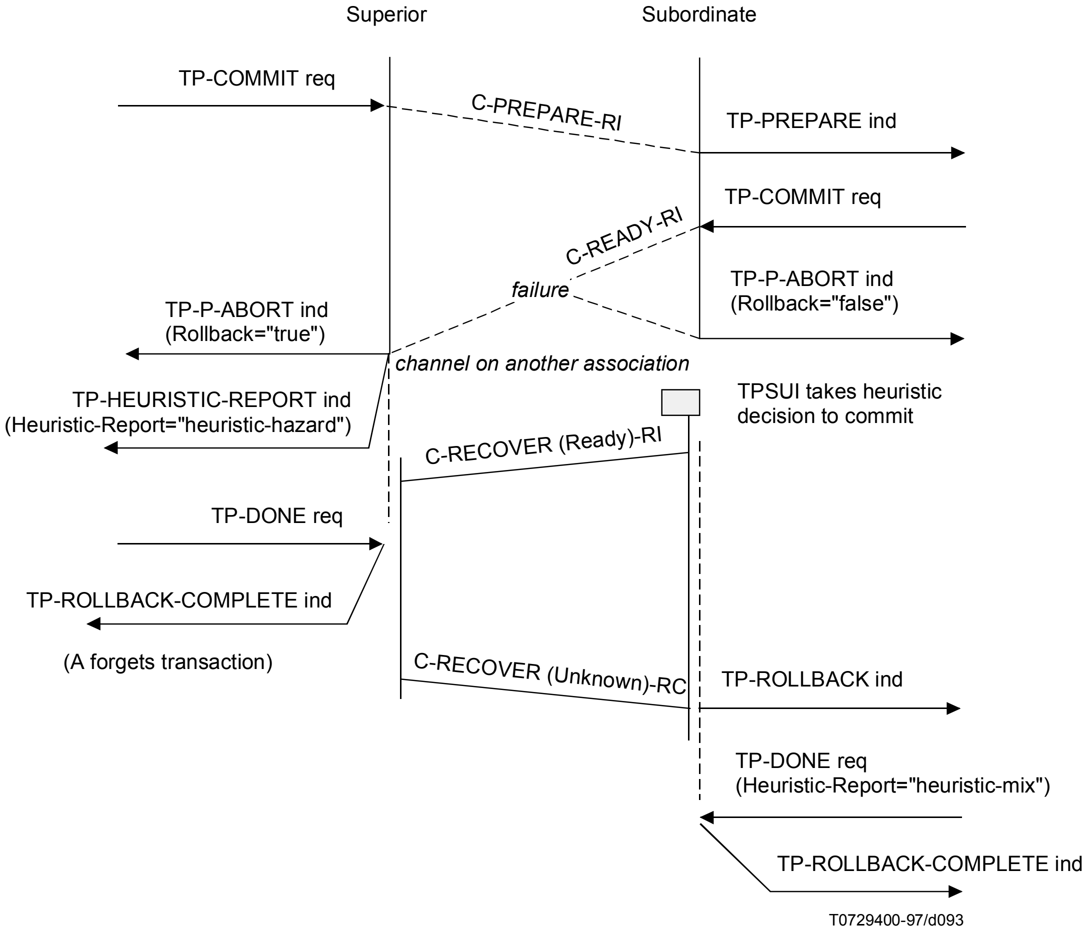


Figure C.88/X.862 – Dialogue aborted during commitment – Subordinate takes (wrong) heuristic decision to roll back

C.8.2.2 Subordinate rolls back – No damage – Heuristic-hazard reported

The scenario of Figure C.89 shows a case where the subordinate decides to heuristically roll back; since the dialogue has failed with the superior before TP-COMMIT indication is issued, the superior rolls back and no recovery occurs, there is no damage but a heuristic-hazard report is given to the superior.

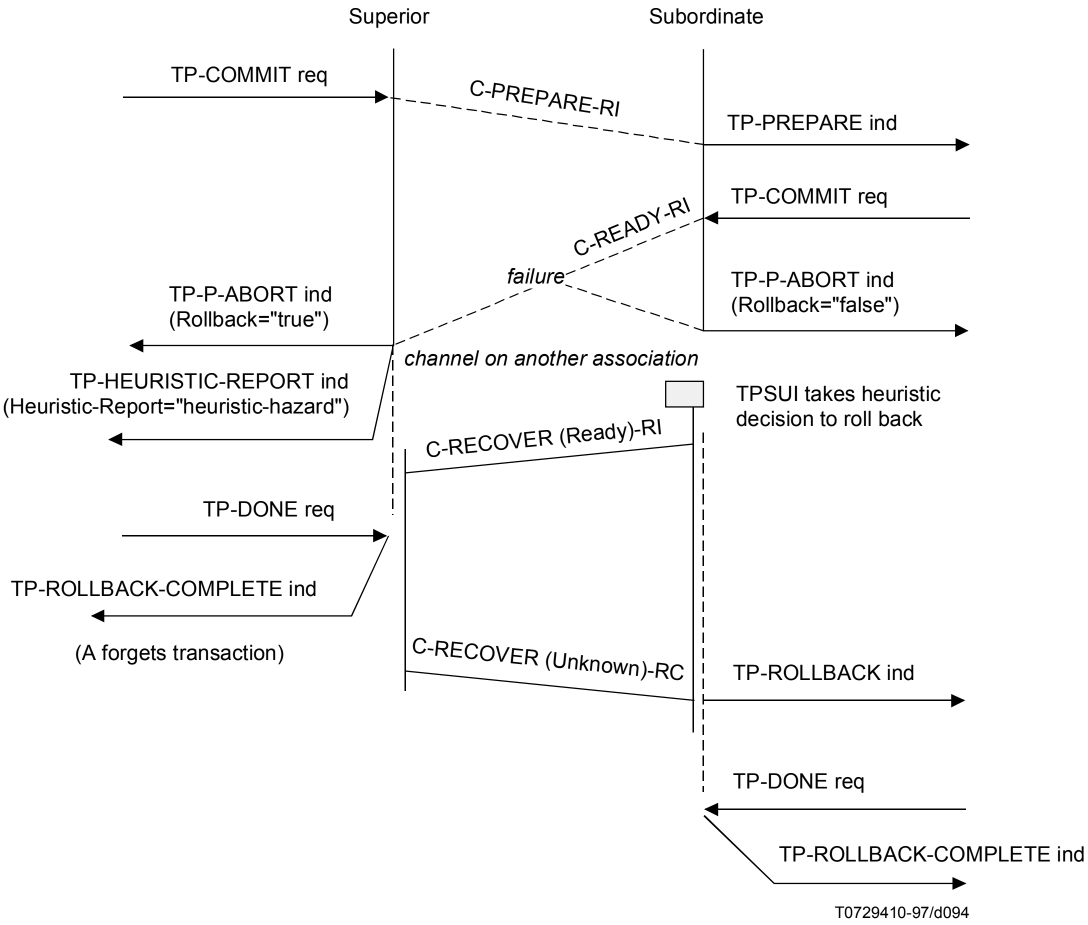


Figure C.89/X.862 – Dialogue aborted during commitment – Subordinate takes (correct) decision to roll back but a heuristic-damage is reported

C.8.2.3 Subordinate rolls back – No damage – Heuristic-hazard reported

The scenario of Figure C.90 shows a case where the subordinate receives a TP-P-ABORT indication with Rollback set to "true" in the active phase. No heuristic decision has been taken but, since the superior has issued TP-COMMIT request, a heuristic-hazard is reported.

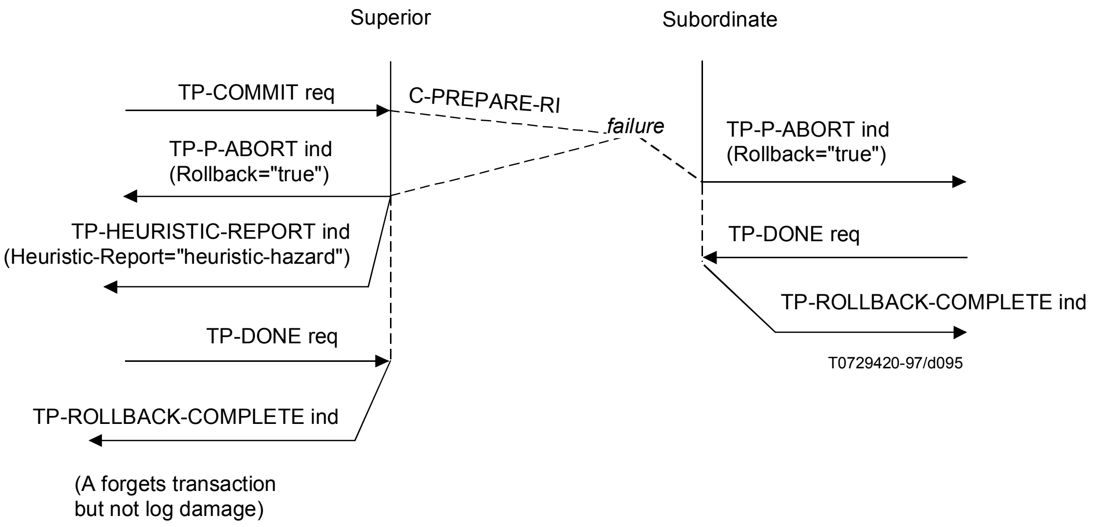


Figure C.90/X.862 – Dialogue aborted before subordinate is aware of transaction termination – No heuristic decision taken but a heuristic-damage is reported

C.8.3 Heuristic decisions in trees with multiple dialogues

The following scenarios show three examples of propagation of heuristic report within a tree having three nodes.

The scenario in Figure C.91 shows a case where the leaf TPSUI decides to heuristically roll back while the outcome of the transaction is commit. The heuristic-mix report is given to the TPSUIs at B node and at A node.

The scenario in Figure C.92 shows a similar case except that the A node is able to compensate and does not issue any heuristic report to its TPSUI.

The scenario in Figure C.93 shows a case where the leaf TPSUI decides to heuristically commit while the outcome of the transaction is rollback. The heuristic-mix report is given to the TPSUIs at B node and A node.

C.9 Scenarios for SACF

The scenario of Figure C.94 shows a case where a contention-loser uses BID to establish a dialogue with provider-supported transaction.

The scenario of Figure C.95 shows a case where a contention-loser does not use BID to establish a dialogue with provider-supported transaction. Figure C.95 is a complicated case where stray APDUs occur.

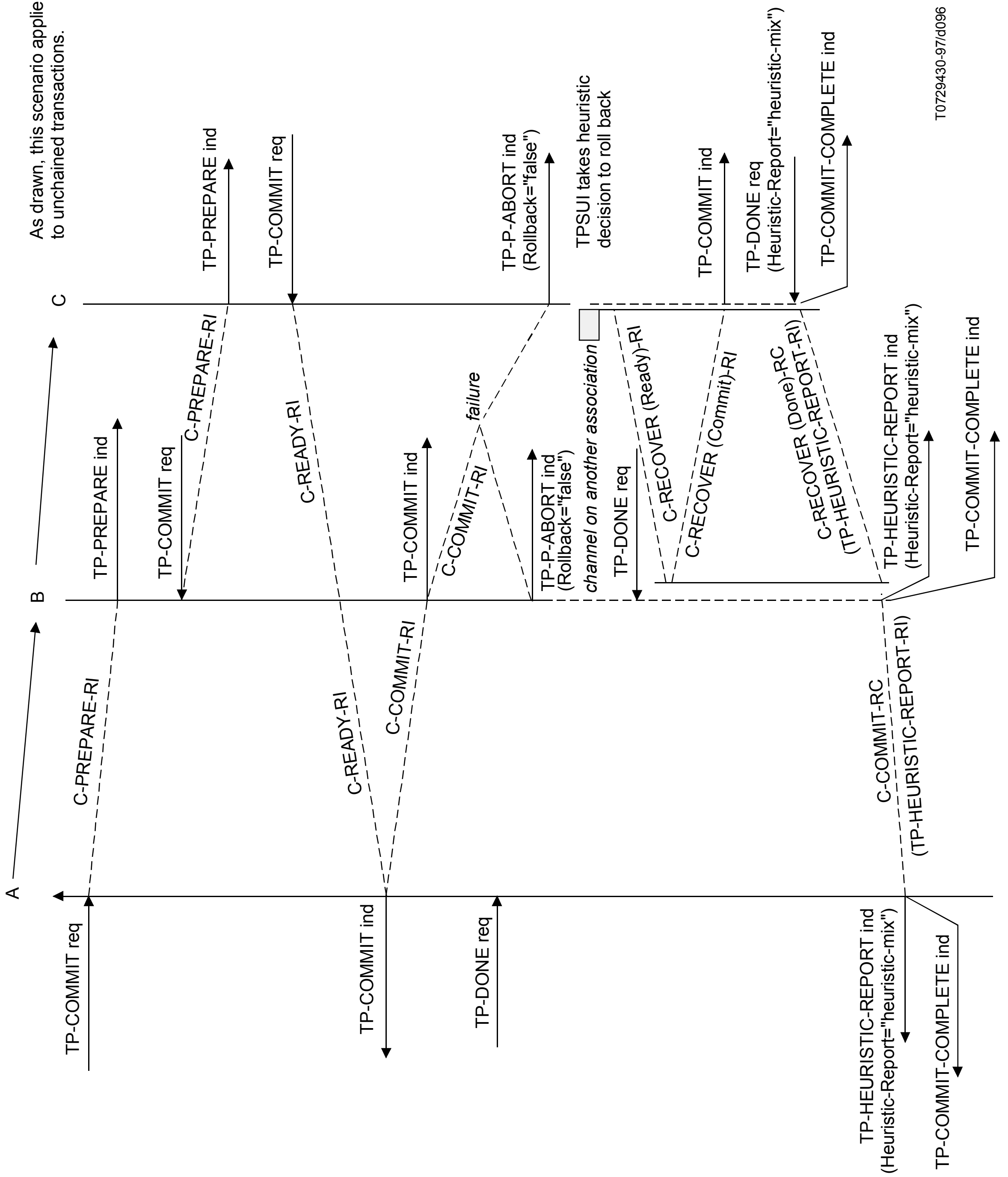

A
B
C
As drawn, this scenario applies to unchained transactions.
TP-COMMIT req
C-PREPARE-RI
TP-PREPARE ind
TP-COMMIT req
C-PREPARE-RI
TP-PREPARE ind
TP-COMMIT req
C-READY-RI
C-READY-RI
TP-COMMIT ind
C-COMMIT-RI
TP-COMMIT ind
C-COMMIT-RI
failure
TP-DONE req
TP-P-ABORT ind (Rollback="false")
TP-P-ABORT ind (Rollback="false")
channel on another association
TPSUI takes heuristic decision to roll back
TP-DONE req
C-RECOVER (Ready)-RI
C-RECOVER (Commit)-RI
TP-COMMIT ind
C-RECOVER (Done)-RC (TP-HEURISTIC-REPORT-RI)
TP-DONE req (Heuristic-Report="heuristic-mix")
TP-COMMIT-COMPLETE ind
TP-HEURISTIC-REPORT ind (Heuristic-Report="heuristic-mix")
TP-COMMIT-COMPLETE ind
C-COMMIT-RC (TP-HEURISTIC-REPORT-RI)
TP-HEURISTIC-REPORT ind (Heuristic-Report="heuristic-mix")
TP-COMMIT-COMPLETE ind
T0729430-97/d096

Figure C.91/X.862 – Heuristic decisions and reporting in a multi-dialogue tree
(All nodes above failure receive heuristic report)

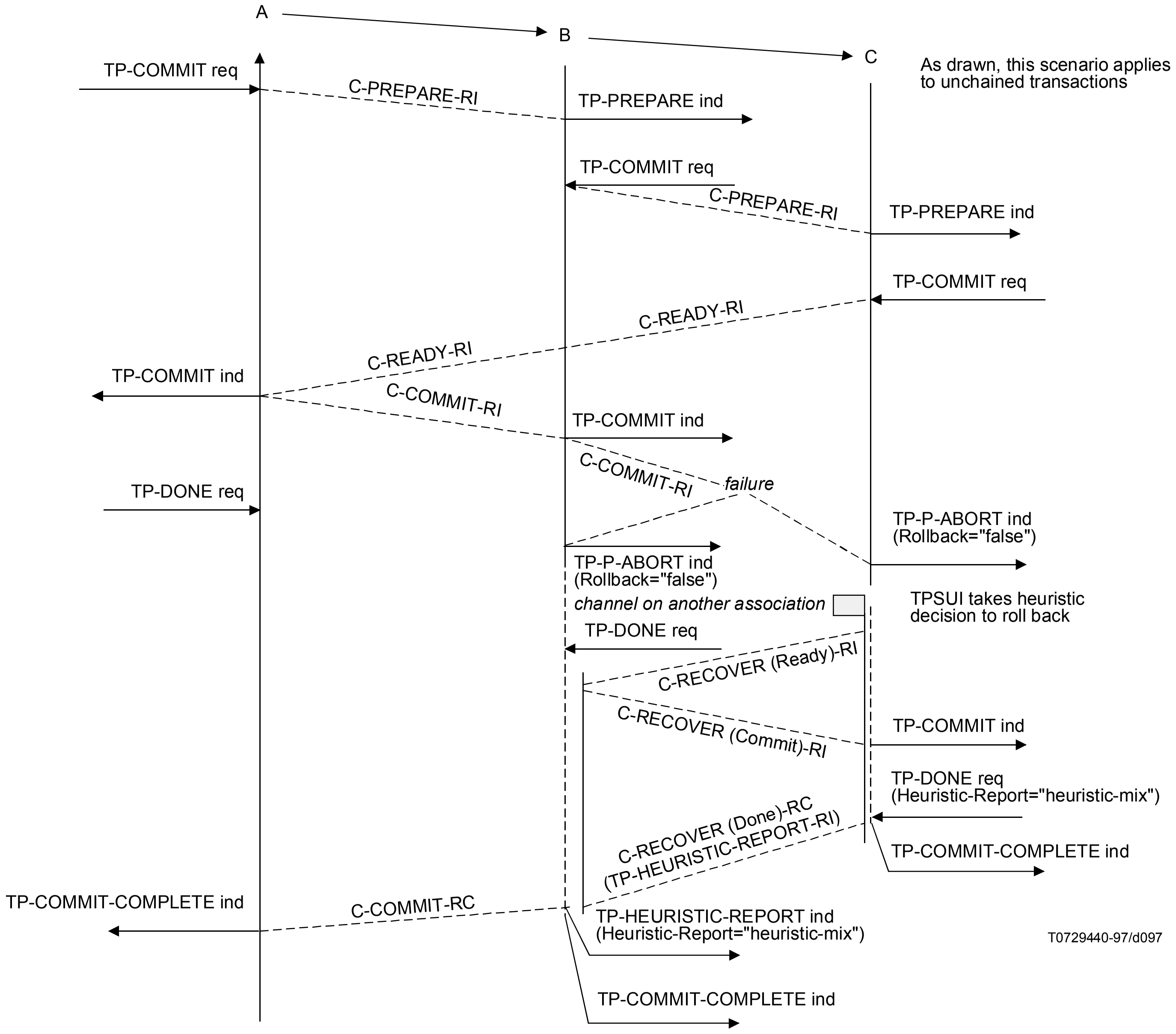

A
B
C
As drawn, this scenario applies to unchained transactions
TP-COMMIT req
C-PREPARE-RI
TP-PREPARE ind
TP-COMMIT req
C-PREPARE-RI
TP-PREPARE ind
TP-COMMIT req
C-READY-RI
C-READY-RI
TP-COMMIT ind
C-COMMIT-RI
TP-COMMIT ind
C-COMMIT-RI
failure
TP-DONE req
TP-P-ABORT ind (Rollback="false")
TP-P-ABORT ind (Rollback="false")
channel on another association
TPSUI takes heuristic decision to roll back
TP-DONE req
C-RECOVER (Ready)-RI
C-RECOVER (Commit)-RI
TP-COMMIT ind
TP-DONE req (Heuristic-Report="heuristic-mix")
C-RECOVER (Done)-RC (TP-HEURISTIC-REPORT-RI)
TP-COMMIT-COMPLETE ind
TP-COMMIT-COMPLETE ind
C-COMMIT-RC
TP-HEURISTIC-REPORT ind (Heuristic-Report="heuristic-mix")
T0729440-97/d097
TP-COMMIT-COMPLETE ind

Figure C.92/X.862 – Heuristic decisions and reporting in a multi-dialogue tree
(Node B compensates before reporting)

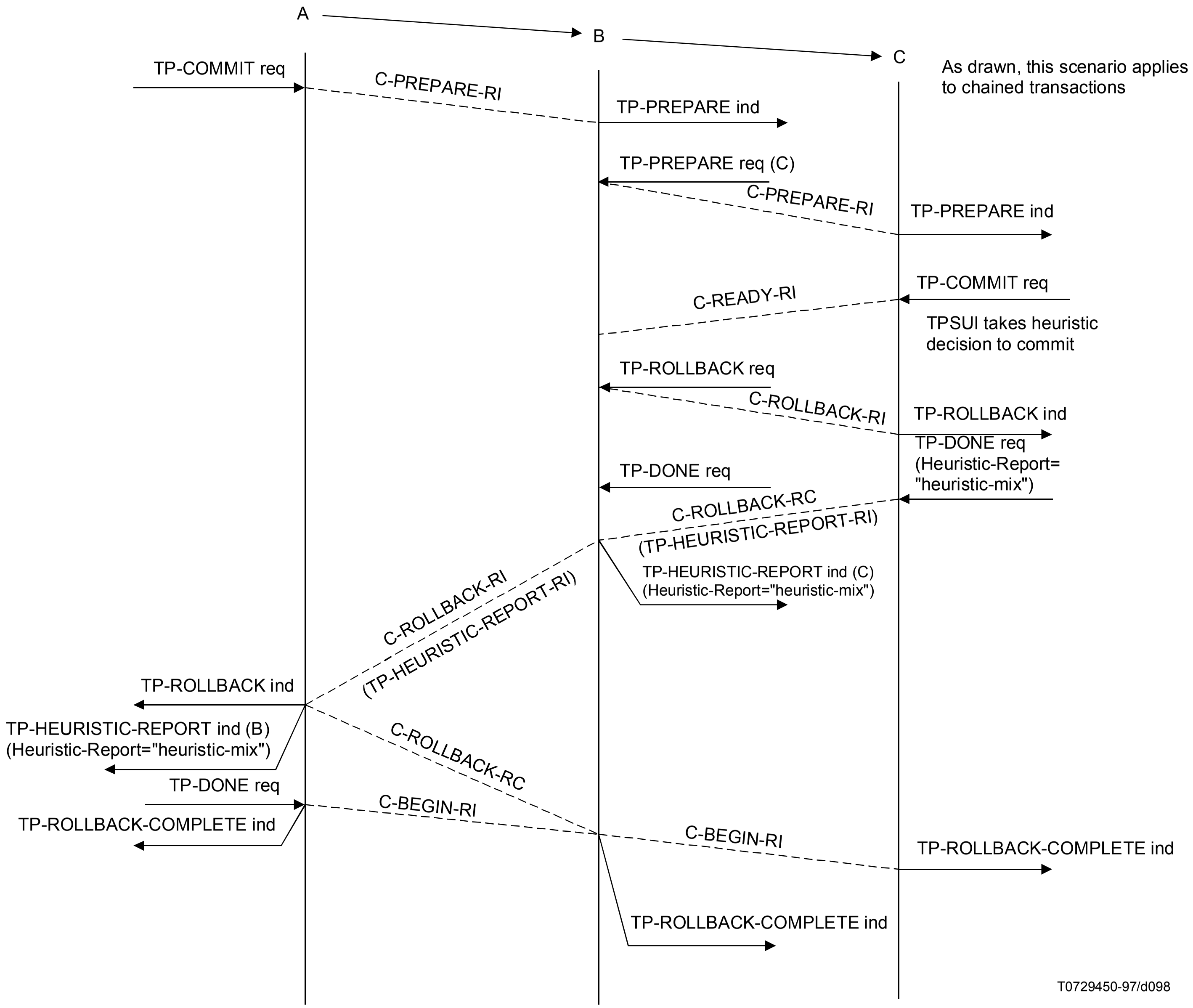
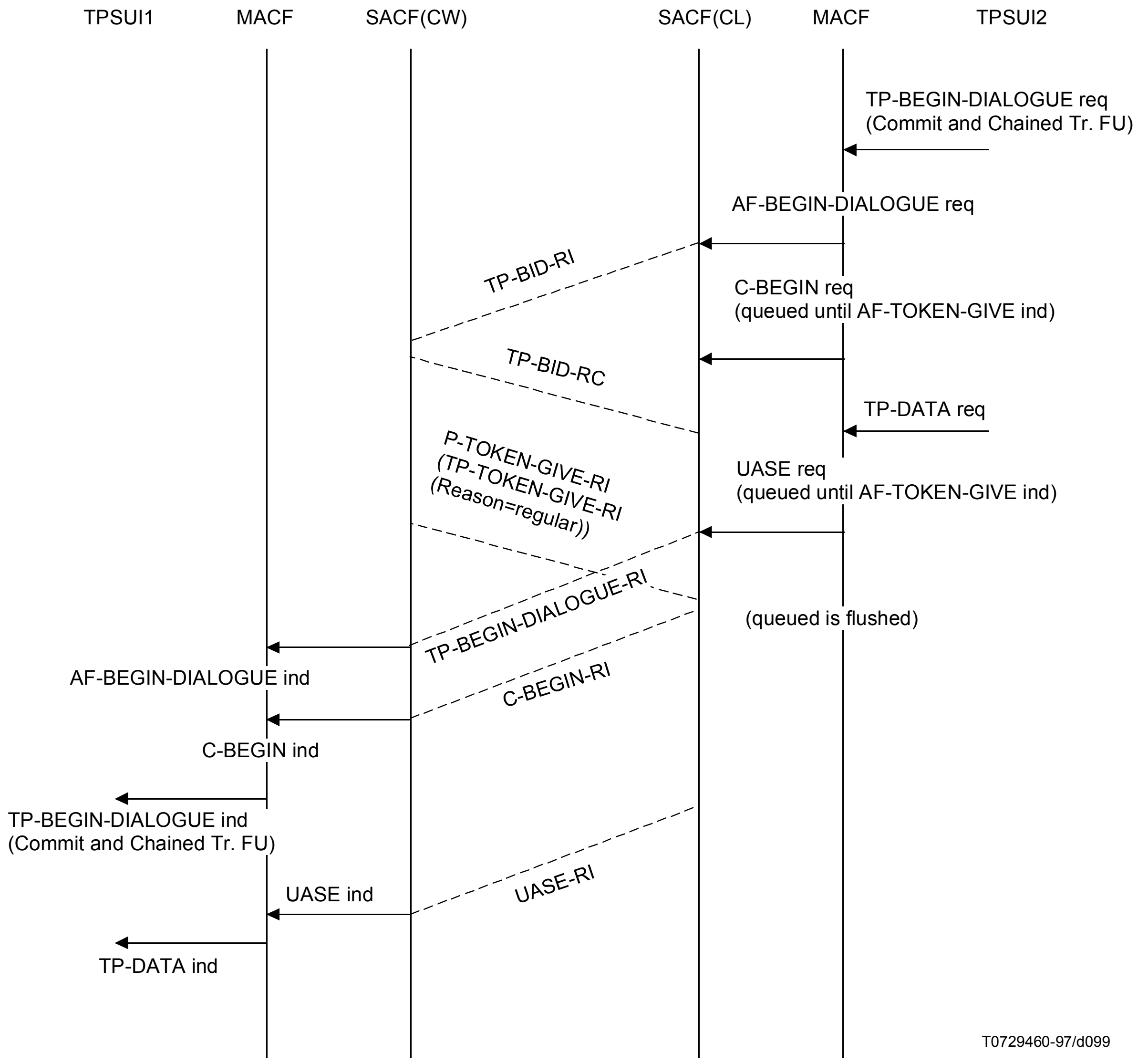

A
B
C
As drawn, this scenario applies to chained transactions
TP-COMMIT req
C-PREPARE-RI
TP-PREPARE ind
TP-PREPARE req (C)
C-PREPARE-RI
TP-PREPARE ind
TP-COMMIT req
C-READY-RI
TPSUI takes heuristic decision to commit
TP-ROLLBACK req
C-ROLLBACK-RI
TP-ROLLBACK ind
TP-DONE req (Heuristic-Report= "heuristic-mix")
TP-DONE req
C-ROLLBACK-RC (TP-HEURISTIC-REPORT-RI)
TP-HEURISTIC-REPORT ind (C) (Heuristic-Report="heuristic-mix")
C-ROLLBACK-RI (TP-HEURISTIC-REPORT-RI)
TP-ROLLBACK ind
TP-HEURISTIC-REPORT ind (B) (Heuristic-Report="heuristic-mix")
C-ROLLBACK-RC
TP-DONE req
C-BEGIN-RI
TP-ROLLBACK-COMPLETE ind
C-BEGIN-RI
TP-ROLLBACK-COMPLETE ind
TP-ROLLBACK-COMPLETE ind
T0729450-97/d098

Figure C.93/X.862 – Heuristic decisions and reporting in a multi-dialogue tree
(heuristic report during rollback procedure)



CW contention-winner
CL contention-loser

Figure C.94/X.862 – BID used in the commit

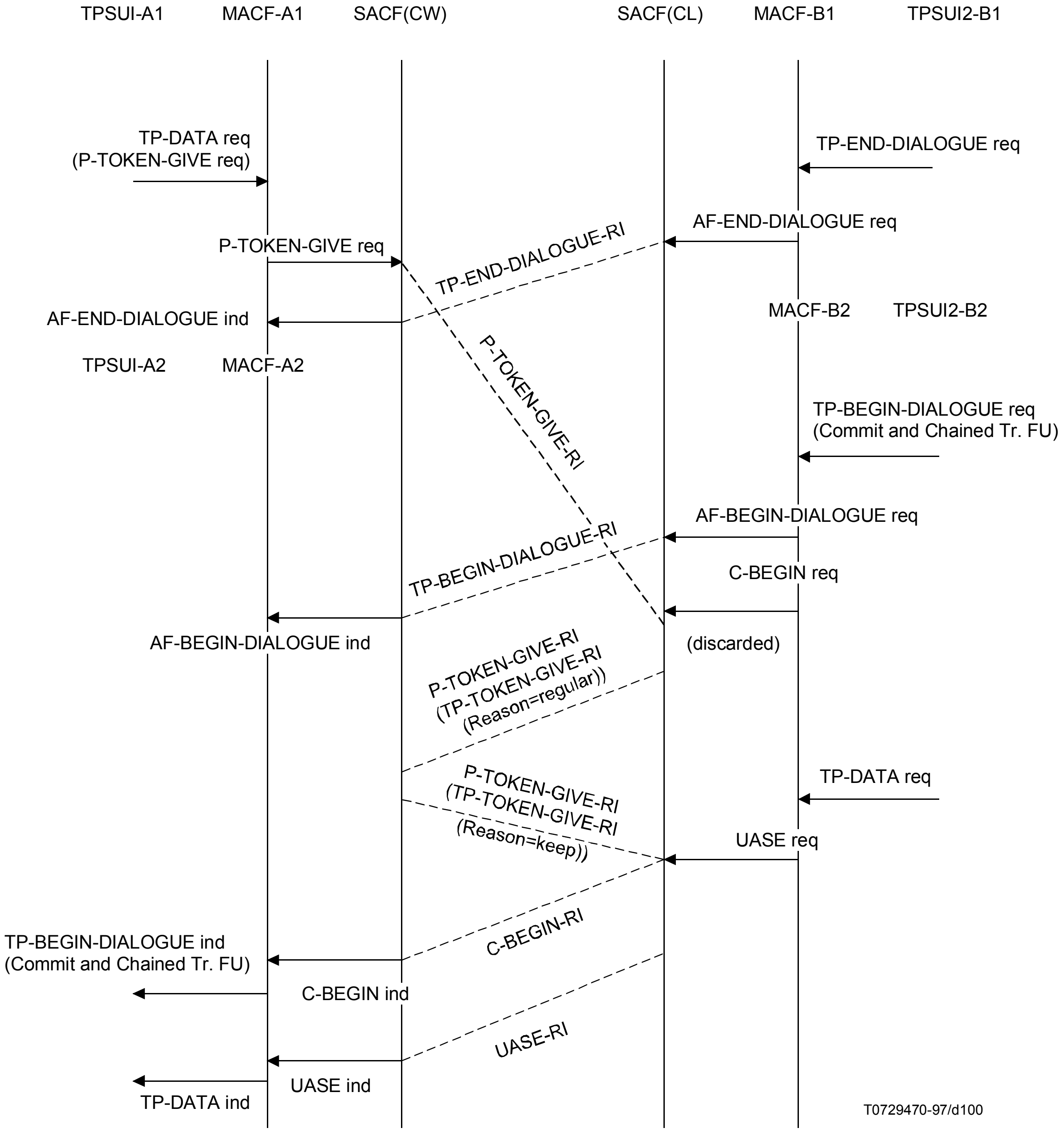


Figure C.95/X.862 – BID not used in the commit

C.10 Scenarios for CPM

C.10.1 Scenarios for channel establishment

The scenario of Figure C.96 shows a simple sequence for channel establishment from a contention-winner.

The scenario of Figure C.97 shows a simple sequence for channel establishment from a contention-loser.

The scenario of Figure C.98 shows a complex sequence for channel establishment from a contention-loser.

In this figure, due to stray APDUs, the token is delayed.

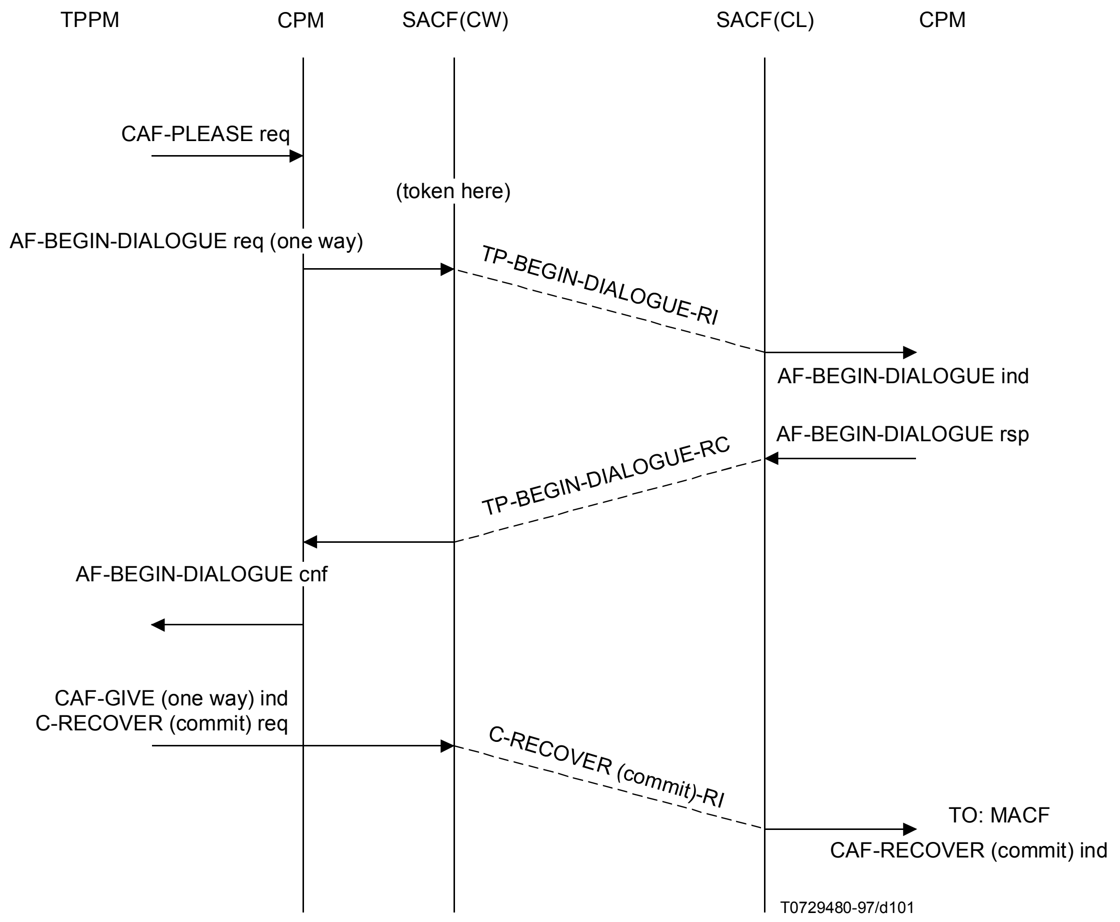


Figure C.96/X.862 – Channel establishment requested by contention-winner (simple case)

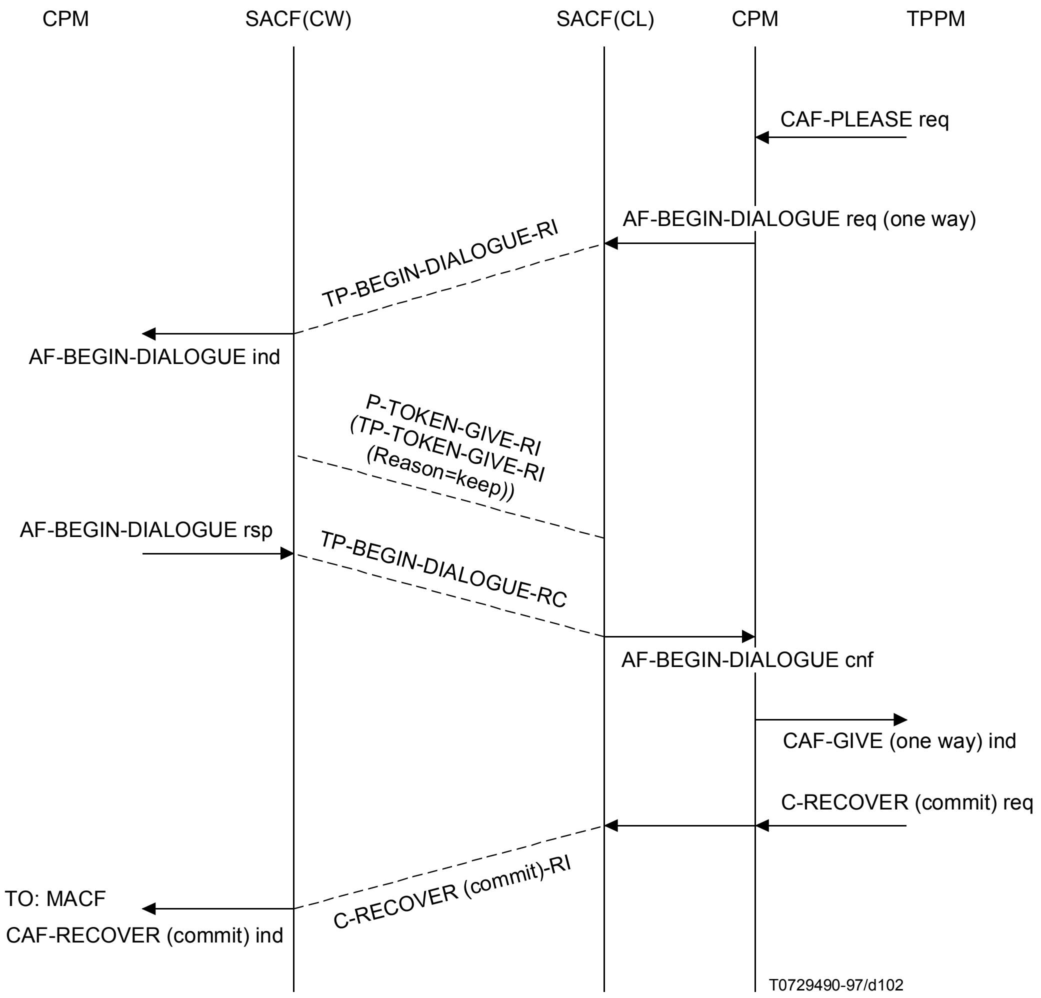


Figure C.97/X.862 – Channel establishment requested by contention-loser (simple case)

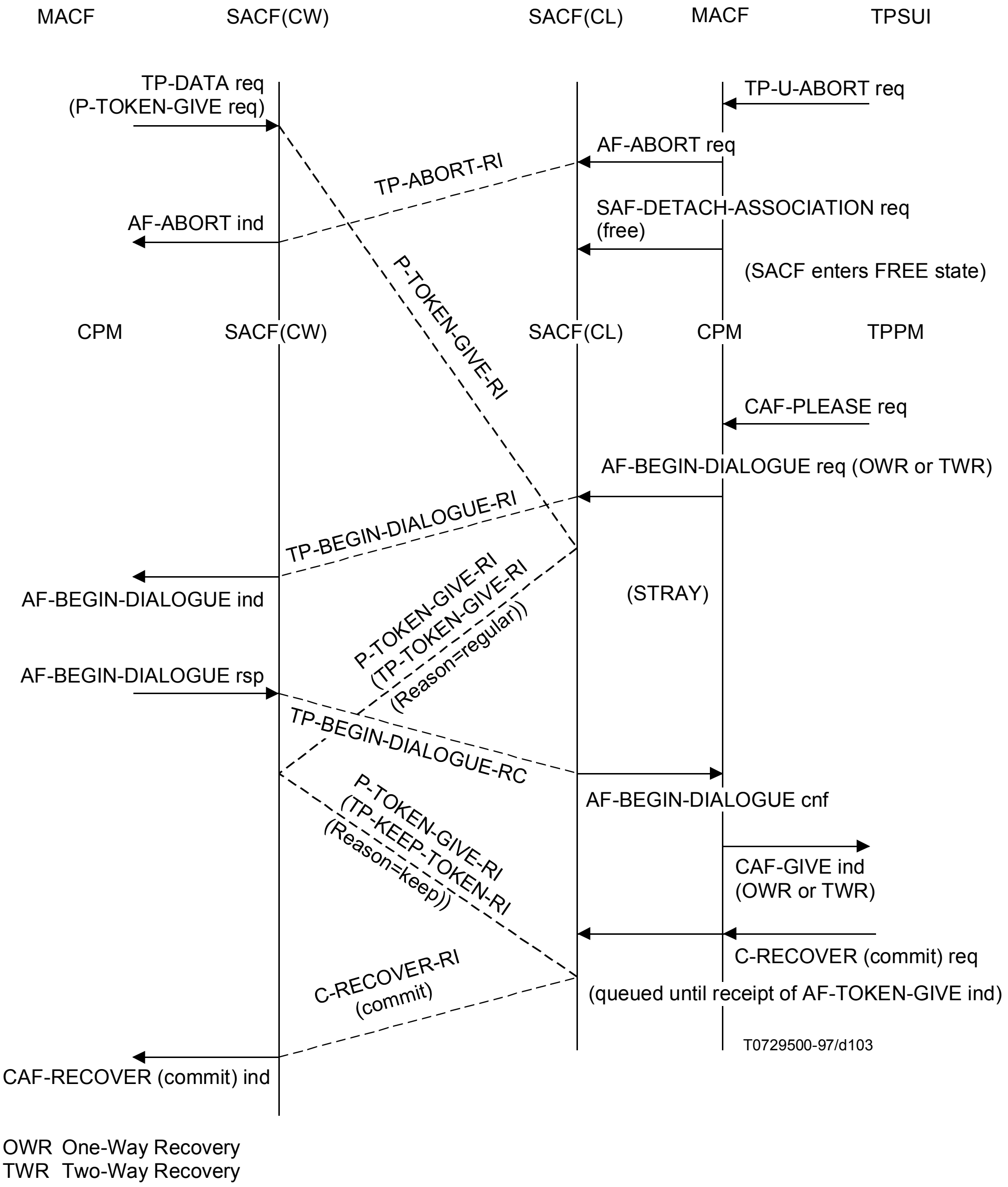


Figure C.98/X.862 – Channel establishment requested by contention-loser (complex case)

C.10.2 Scenarios for two way recovery

The scenario of Figure C.99 shows some sequences for two way recovery.

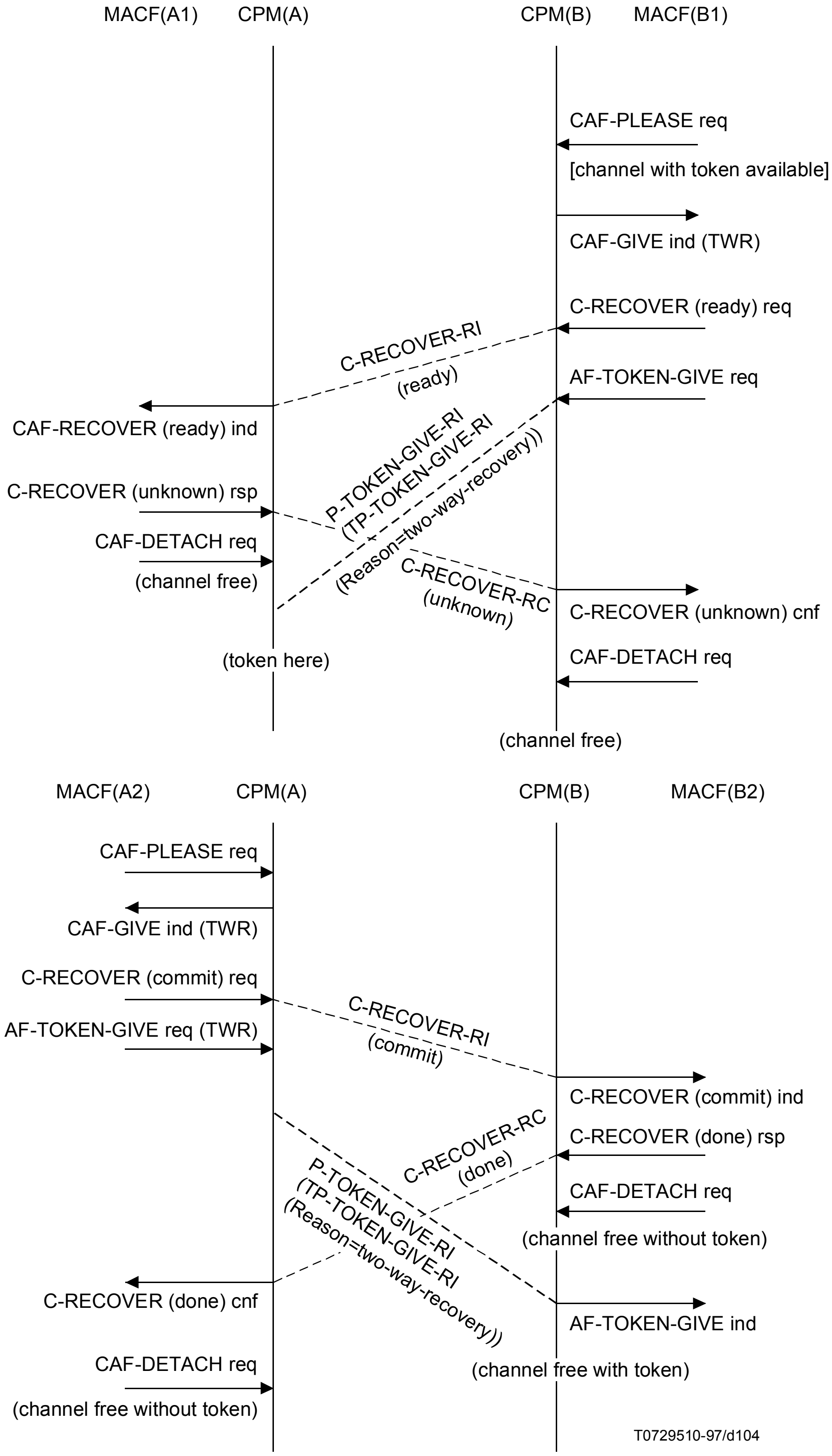


Figure C.99/X.862 – Two way recovery (sheet 1 of 2)

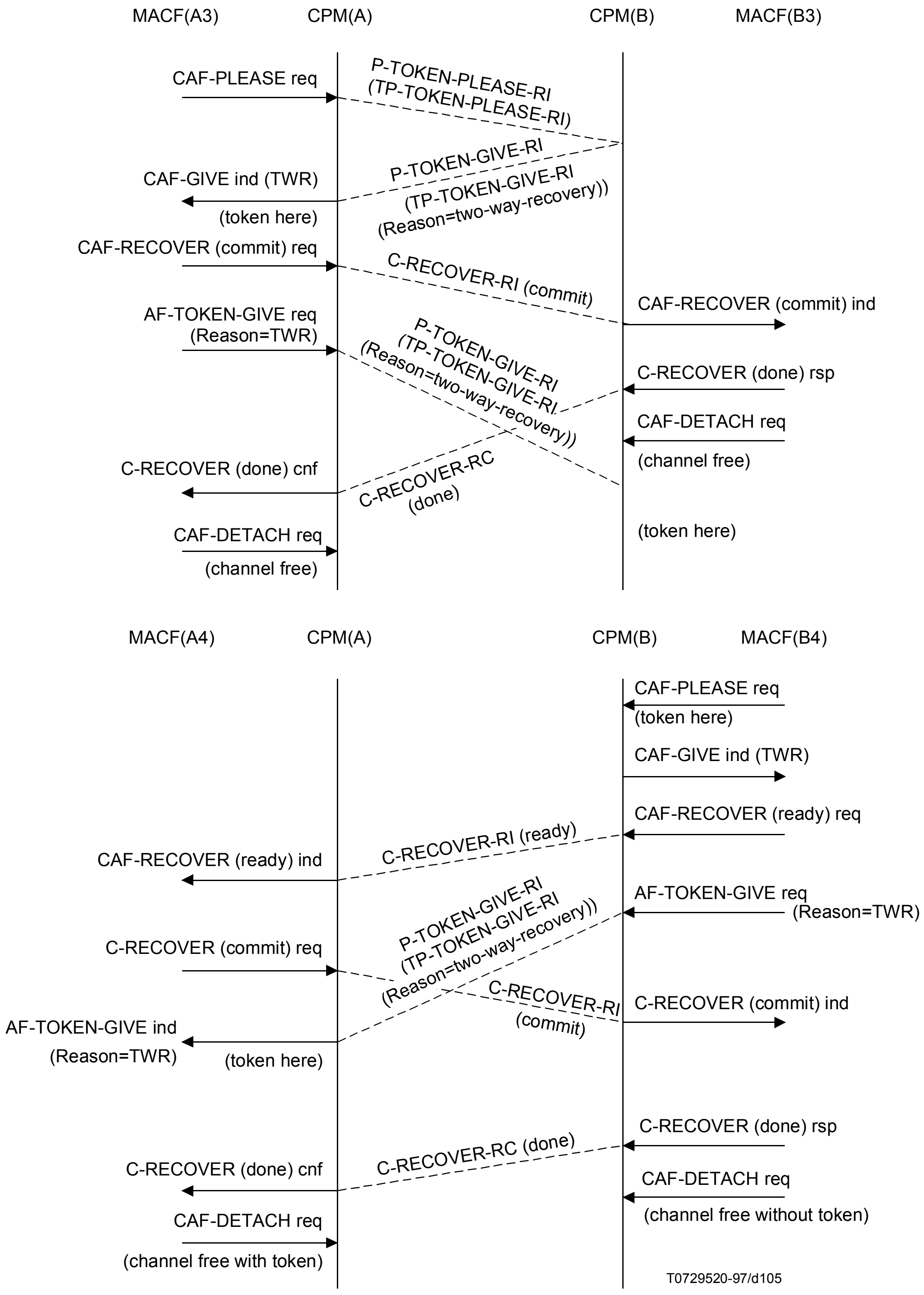


Figure C.99/X.862 – Two way recovery (sheet 2 of 2)

C.10.3 Scenarios for recovery collision on separate one way channels

The scenario of Figure C.100 shows a case when recovery collision occurs because both a superior and a subordinate initiate recovery. This case can occur when, for example, C-READY-RI or C-COMMIT-RI cannot reach the partner MACF due to the communication failure. When recovery collision occurs, the superior always issues C-RECOVER (retry-later) response as a response to CAF-RECOVER (ready) indication and the subordinate always issues C-RECOVER (done) response as a response to CAF-RECOVER (commit) indication.

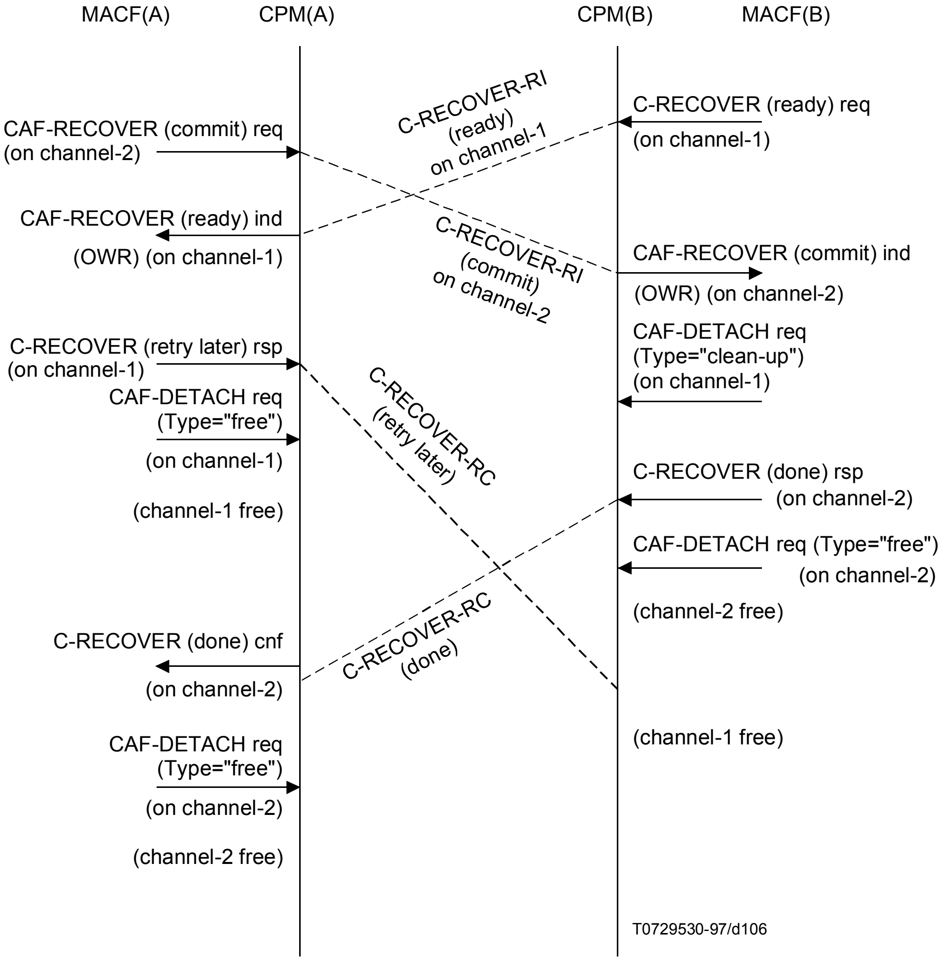


Figure C.100/X.862 – Recovery collision due to requests by both a superior and a subordinate

C.11 Read-Only scenarios

C.11.1 TP-READ-ONLY request with unchained transactions

The scenario of Figure C.101 shows the use of the Read-Only functional unit with the Commit and Unchained Transactions functional units. At the end of this scenario, the coordination level of the dialogue is none and the subordinate is no longer involved in the transaction. The superior will continue with the transaction and whether the transaction is finally committed or rolled back will not affect activity related to the dialogue.

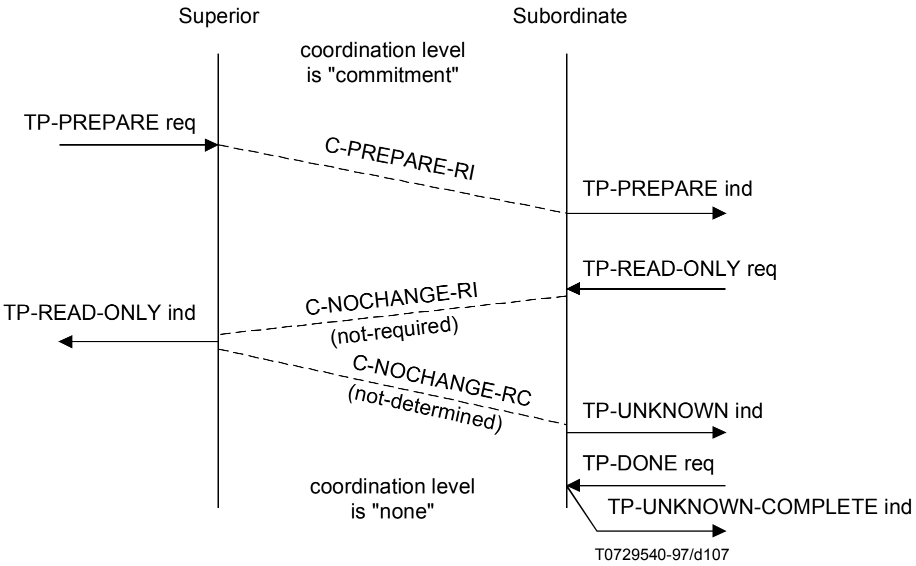


Figure C.101/X.862 – TP-READ-ONLY request with unchained transactions

C.11.2 TP-READ-ONLY request with chained transactions – Transaction commits

The scenario of Figure C.102 shows the use of the Read-Only functional unit with the Commit and Chained Transactions functional units. On receiving the TP-READ-ONLY indication, the superior proceeds to commitment; the dialogue to the subordinate is blocked awaiting the C-BEGIN-RI for the next transaction in the chain. Note that in this case, C-BEGIN-RI is sent as late as possible to minimize the chances of a failure requiring the next transaction at the superior to roll back.

From an OSI TP perspective, there is no requirement for a commit log at the superior because there is no obligation to perform recovery; however there must be a local equivalent of a commit log to ensure that the ACID properties are not violated at the superior.

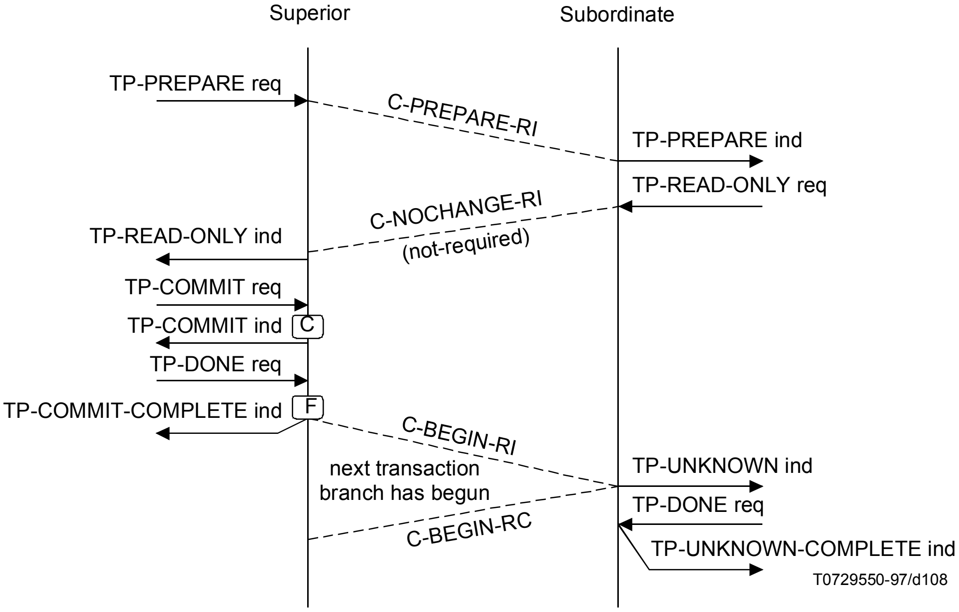


Figure C.102/X.862 – TP-READ-ONLY request with chained transactions – Transaction commits

C.11.3 TP-READ-ONLY request with chained transactions – Transaction is rolled back

The scenario of Figure C.103 shows the use of the Read-Only functional unit with the Commit and Chained Transactions functional units. After receiving the TP-READ-ONLY indication, the superior rolls back the transaction. The Polarized Control case is shown.

If the superior had issued TP-ROLLBACK request before receiving the TP-READ-ONLY indication, the C-ROLLBACK-RI would have purged the C-NOCHANGE-RI; the service sequence at the subordinate would be identical to that shown.

The TP-ROLLBACK-COMPLETE indications indicate that control is with the superior.

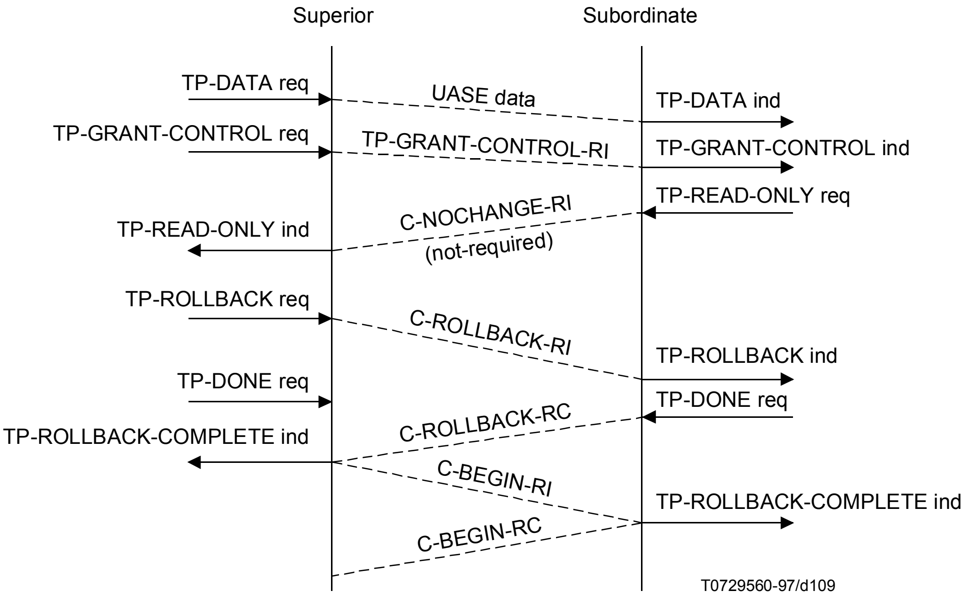


Figure C.103/X.862 – TP-READ-ONLY request with chained transactions – Transaction is rolled back

C.11.4 Dialogue abort after TP-READ-ONLY indication – Chained transactions

The case shown in Figure C.104 is for chained transactions. The dialogue abort is detected before the C-BEGIN-RI is sent for the next transaction.

For the unchained transactions case, a C-NOCHANGE-RI (not-required) would be sent, responded to by a C-NOCHANGE-RC (not-determined); if the abort killed the response, the service events would be as below.

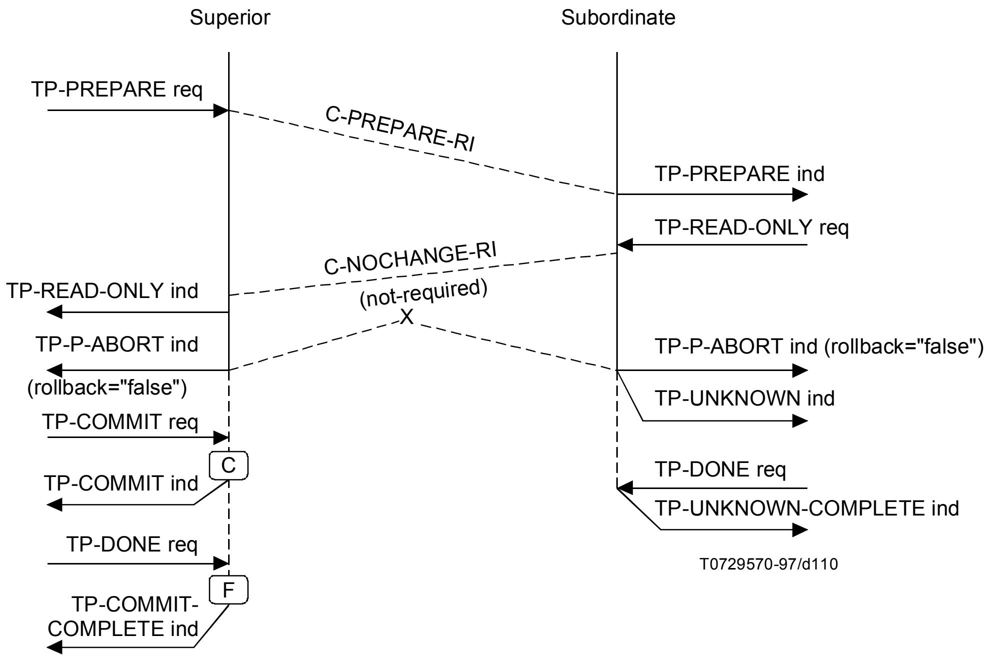


Figure C.104/X.862 – Dialogue abort after TP-READ-ONLY indication – Chained transactions

C.11.5 Dialogue abort before TP-READ-ONLY indication – Chained transactions

The case shown in Figure C.105 is for chained transactions. The dialogue abort kills the C-NOCHANGE-RI and the transaction is rolled back at the superior; the events at the subordinate are identical to the previous scenario.

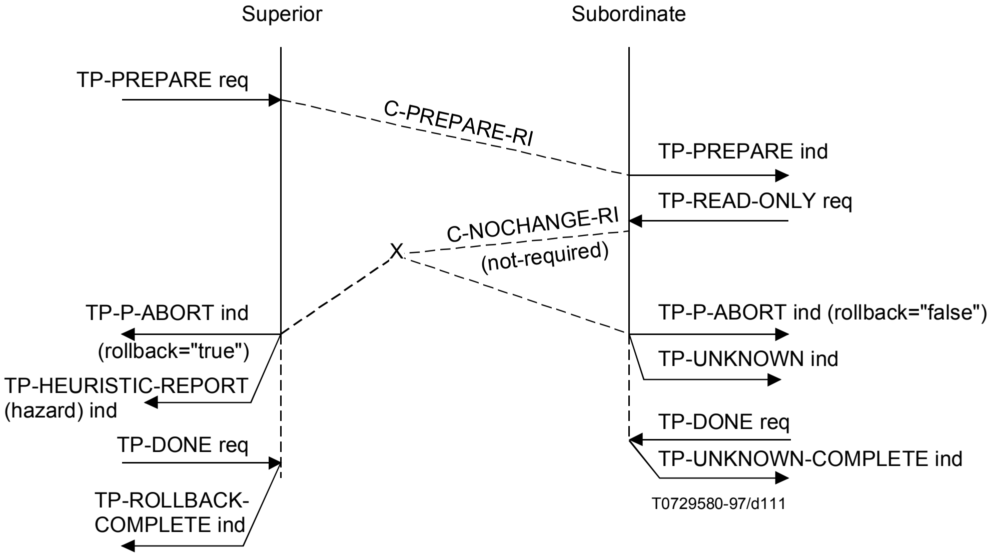


Figure C.105/X.862 – Dialogue abort before TP-READ-ONLY indication – Chained transactions

C.11.6 User dialogue abort after TP-READ-ONLY indication – Chained transactions

The case shown in Figure C.106 is for chained transactions. The TP-U-ABORT request is for a dialogue still in the transaction which will not be rolled back as clearly there is no threat to bound data.

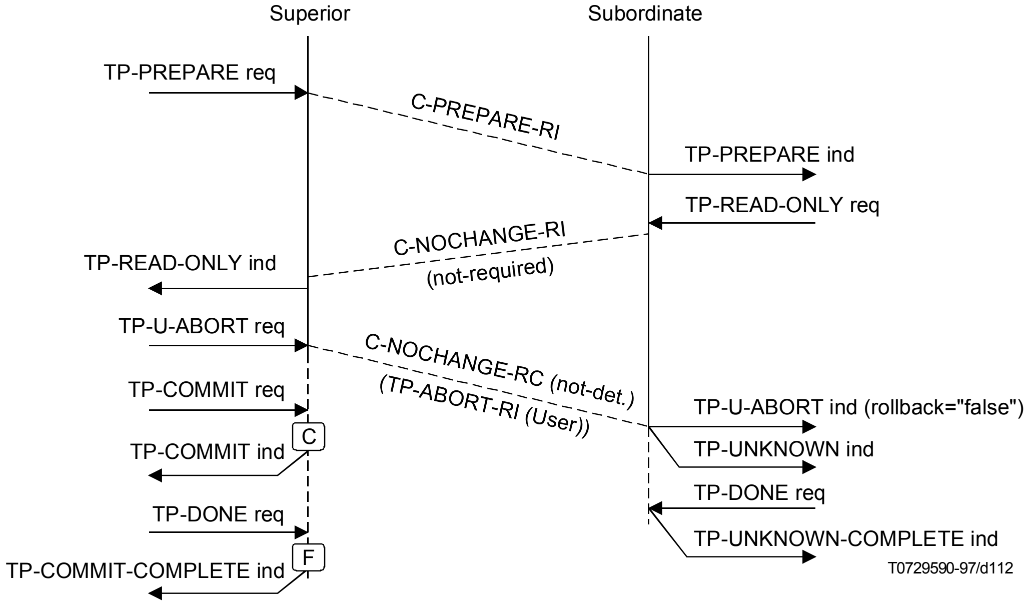


Figure C.106/X.862 – User dialogue abort after TP-READ-ONLY indication – Chained transactions

C.11.7 TP-READ-ONLY request at intermediate and leaf – Chained transactions

TP-READ-ONLY request invoked at an intermediate and leaf nodes. The Chained Transactions case is shown in Figure C.107.

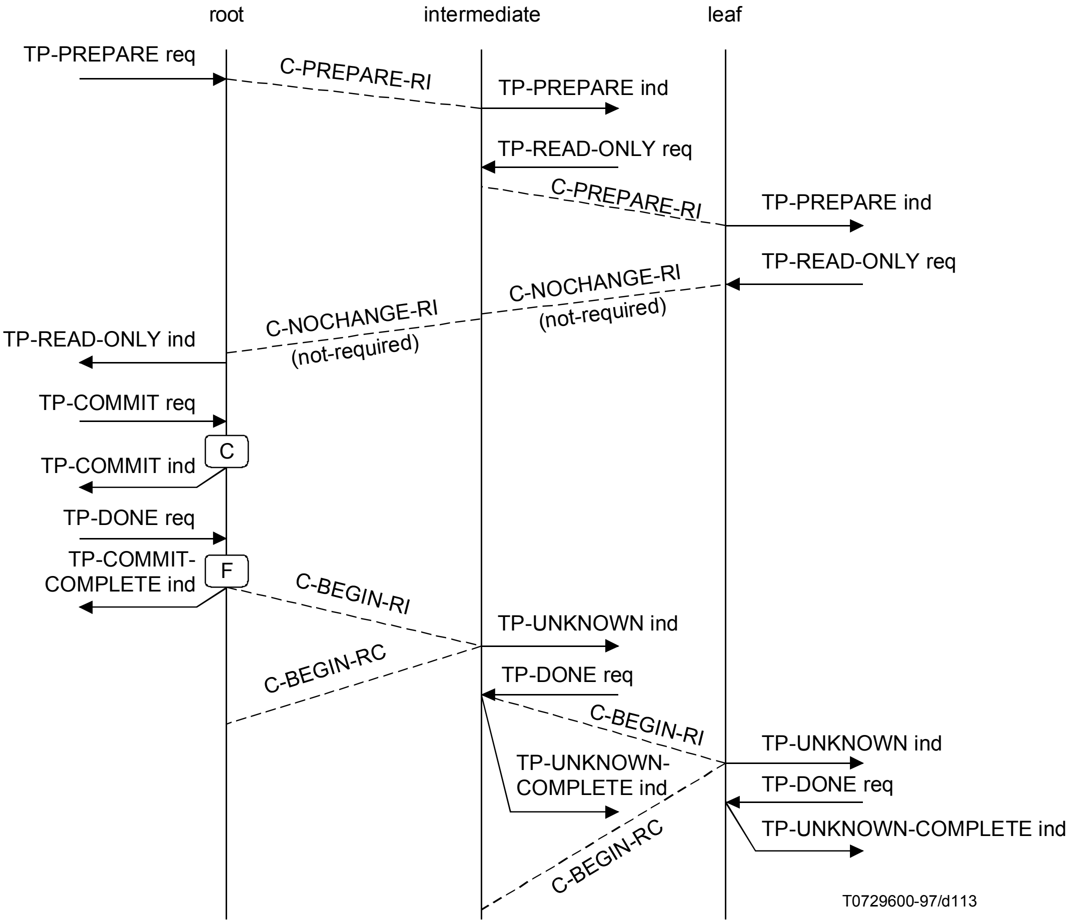


Figure C.107/X.862 – TP-READ-ONLY request at intermediate and leaf – Chained transactions

C.11.8 TP-READ-ONLY request at intermediate – Leaf issues TP-COMMIT request

TP-READ-ONLY request invoked at an intermediate node; however the leaf node invokes TP-COMMIT request and so full two-phase commit procedures are invoked. As a result, the intermediate node is informed of the transaction outcome and must write a ready log. The Unchained Transactions case is shown in Figure C.108.

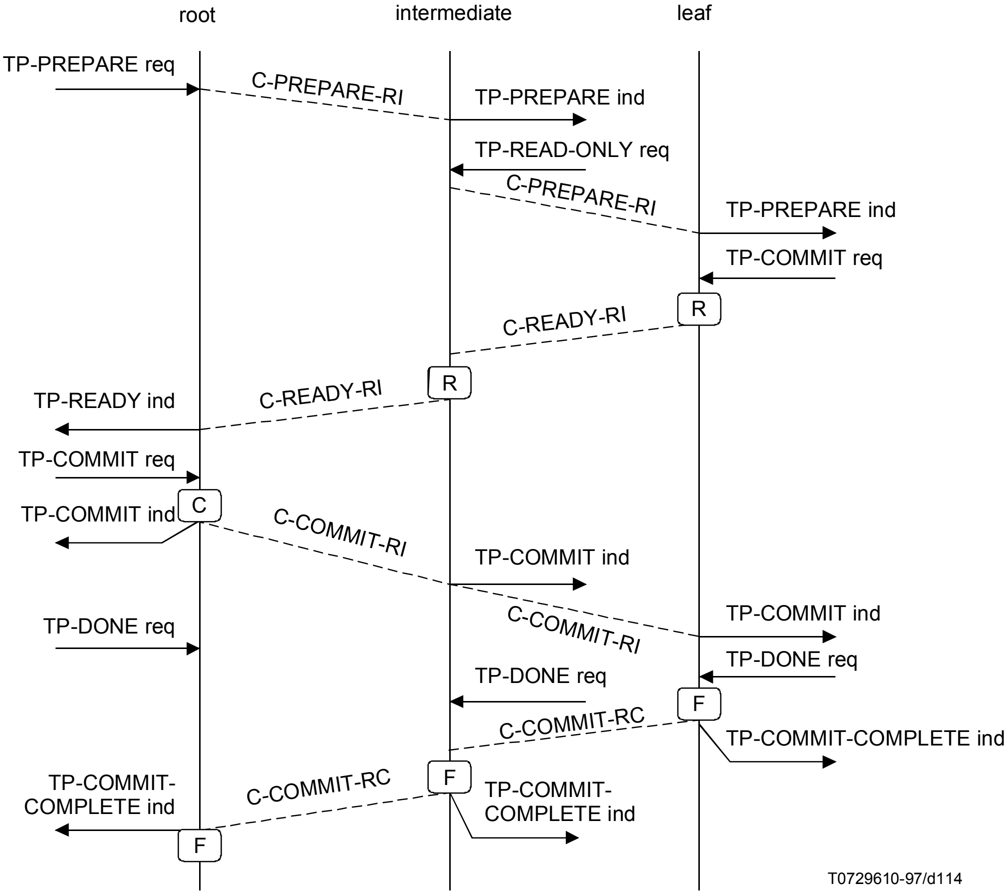


Figure C.108/X.862 – TP-READ-ONLY request at intermediate – Leaf issues TP-COMMIT request

C.11.9 TP-READ-ONLY request at intermediate – Leaf issues TP-ROLLBACK request

TP-READ-ONLY request invoked at an intermediate node; however the leaf node invokes TP-ROLLBACK request and so the full rollback procedure is invoked. As a result, the intermediate node is informed of the transaction outcome. The Unchained Transactions case is shown in Figure C.109.

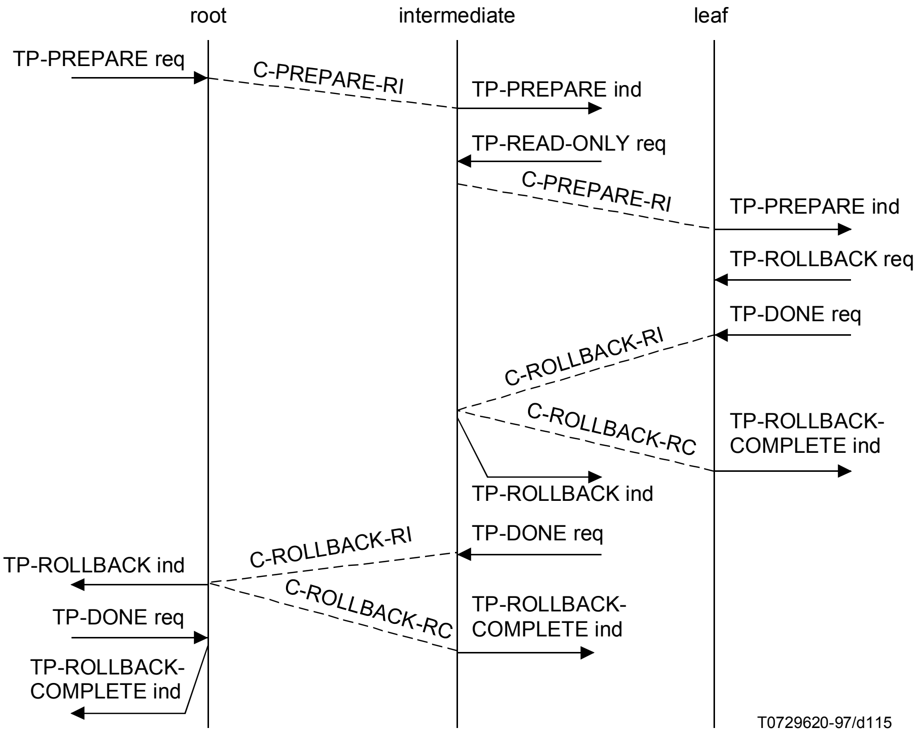


Figure C.109/X.862 – TP-READ-ONLY request at intermediate – Leaf issues TP-ROLLBACK request

C.11.10 TP-READ-ONLY request at intermediate – Leaf issues TP-ROLLBACK request and diagnostics

TP-READ-ONLY request invoked at an intermediate node; however the leaf node invokes TP-ROLLBACK request and so the full rollback procedure is invoked; the Completion Diagnostics functional unit is selected on both dialogues. As a result, the intermediate node is informed of the transaction outcome, and receives a completion report. The Unchained Transactions case is shown in Figure C.110.

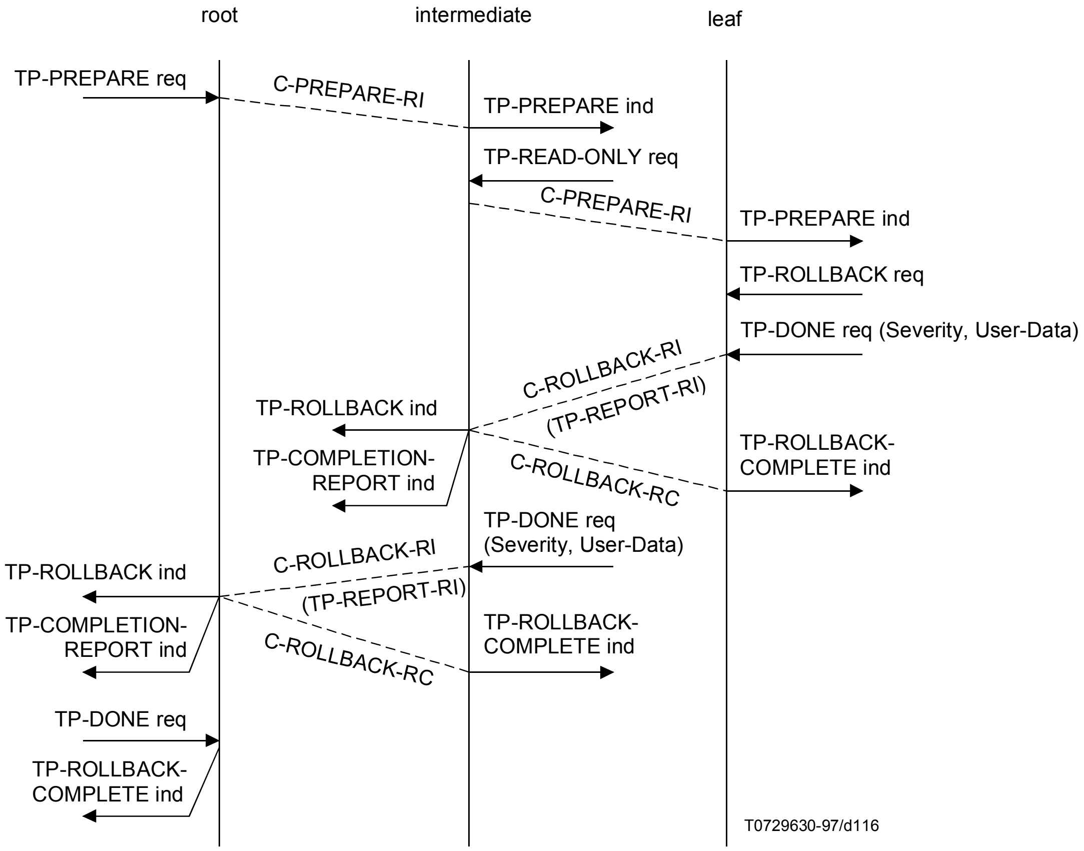


Figure C.110/X.862 – TP-READ-ONLY request at intermediate – Leaf issues TP-ROLLBACK request and diagnostics

C.11.11 TP-READ-ONLY request at intermediate – Leaf issues TP-COMMIT request but heuristically rolls back

TP-READ-ONLY request invoked at an intermediate node; however the leaf node invokes TP-COMMIT request and so full two-phase commit procedures are invoked. The leaf node makes a heuristic decision to roll back which proves to be incorrect. As a result, the intermediate node is informed of the transaction outcome and receives a heuristic report, which is propagated to the root. The Unchained Transactions case is shown in Figure C.111.

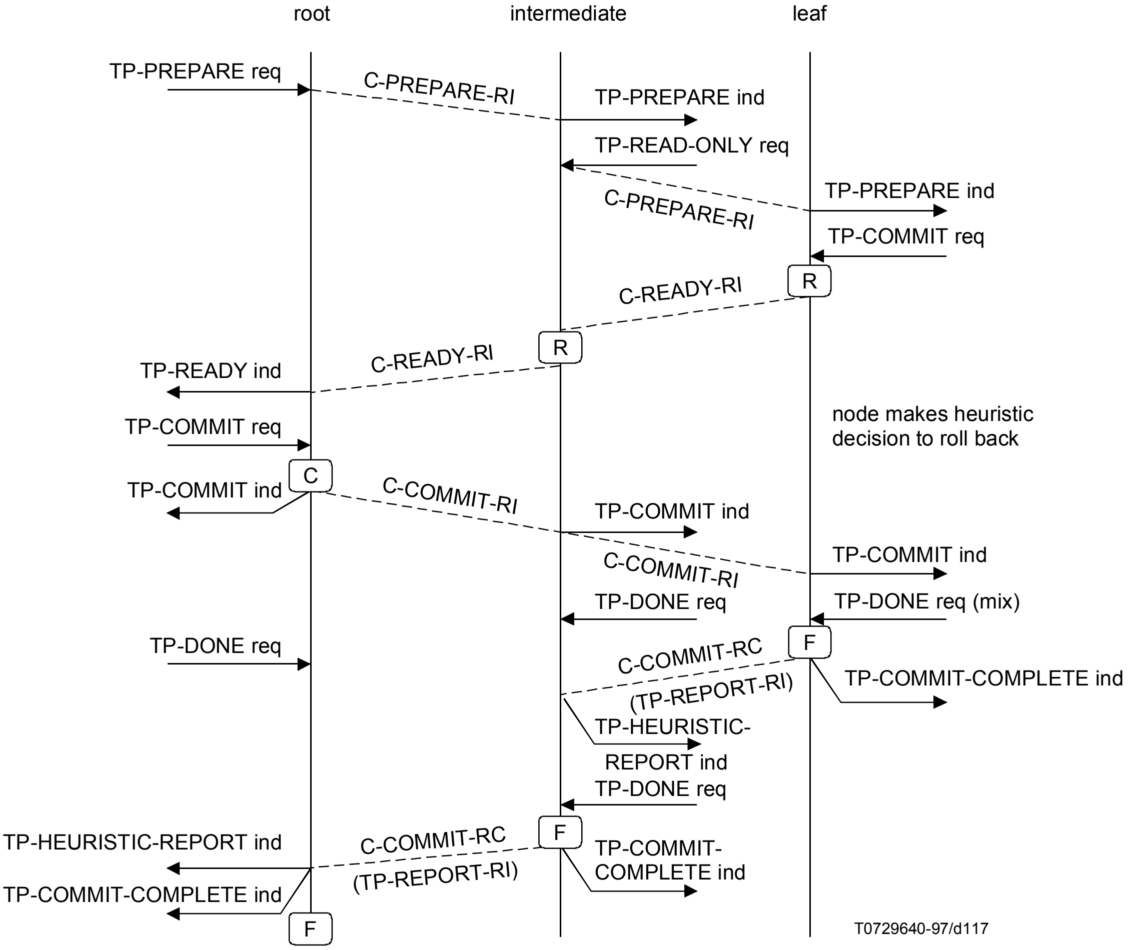


Figure C.111/X.862 – TP-READ-ONLY request at intermediate – Leaf issues TP-COMMIT request but heuristically rolls back

C.11.12 TP-READ-ONLY request with deferred end dialogue – Transaction commits

The scenario of Figure C.112 shows a leaf C issuing TP-READ-ONLY request, but the deferred end dialogue request causes the reply to be delayed until the result is known not to be rollback.

In the case of an outstanding defer as soon as result is known not to be rollback, a C-NOCHANGE-RC (Atomic Action Outcome = "not-determined") is sent even if the outcome is known to be commit, so a TP-UNKNOWN indication is signalled at C even though the result is known at node B.

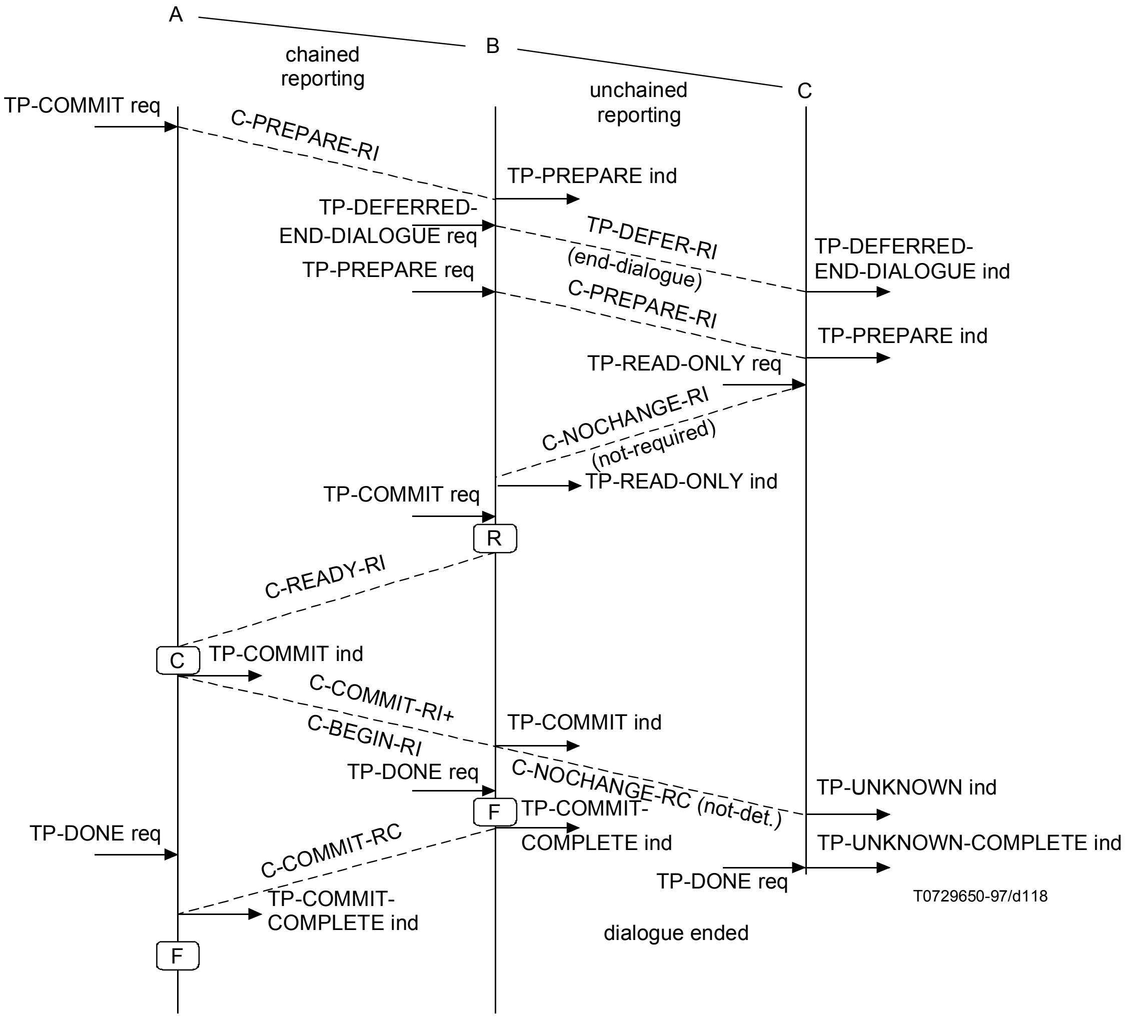


Figure C.112/X.862 – TP-READ-ONLY request with deferred end dialogue – Transaction commits

C.11.13 TP-READ-ONLY request with deferred end dialogue – Transaction rolls back

This is a similar case but the transaction is rolled back by the root TPSUI; the full rollback procedures are used on all dialogues.

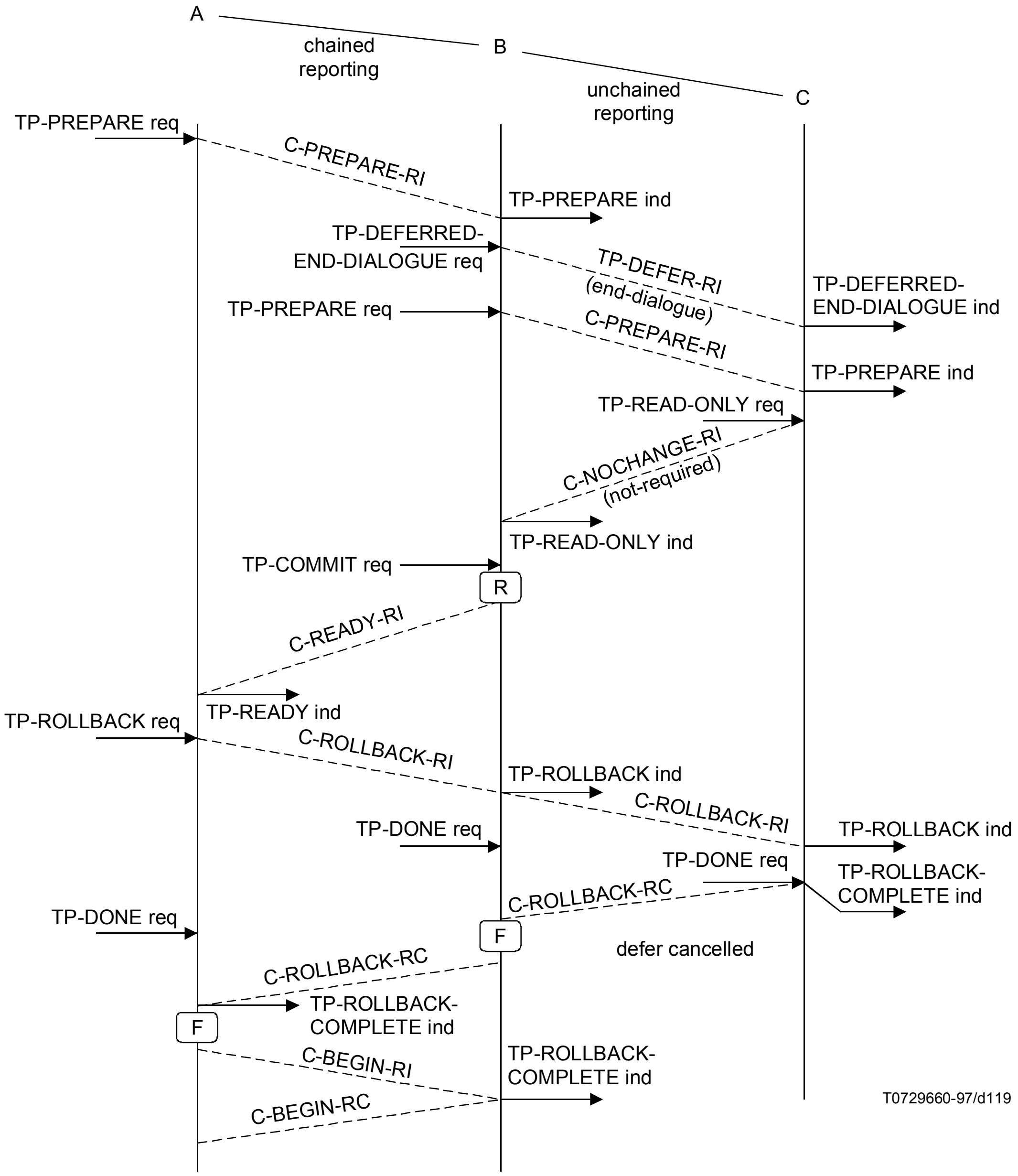


Figure C.113/X.862 – TP-READ-ONLY request with deferred end dialogue – Transaction rolls back

C.11.14 Read-Only Intermediate promoted to Root Node

The scenario of Figure C.114 shows the intermediate node being promoted to root node following a dialogue failure. Chained transactions.

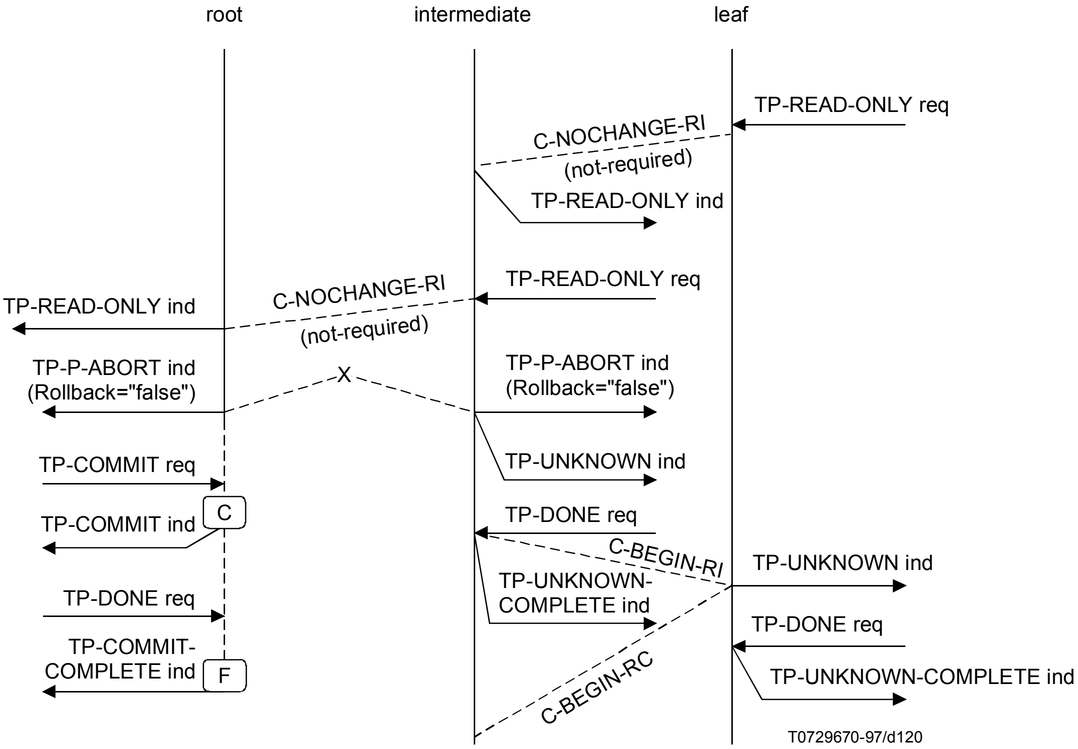


Figure C.114/X.862 – Read-Only Intermediate promoted to Root Node

C.11.15 Read-Only Intermediate refuses to become Root Node

The scenario of Figure C.115 shows the intermediate node refusing to become root node following a dialogue failure. Chained transactions.

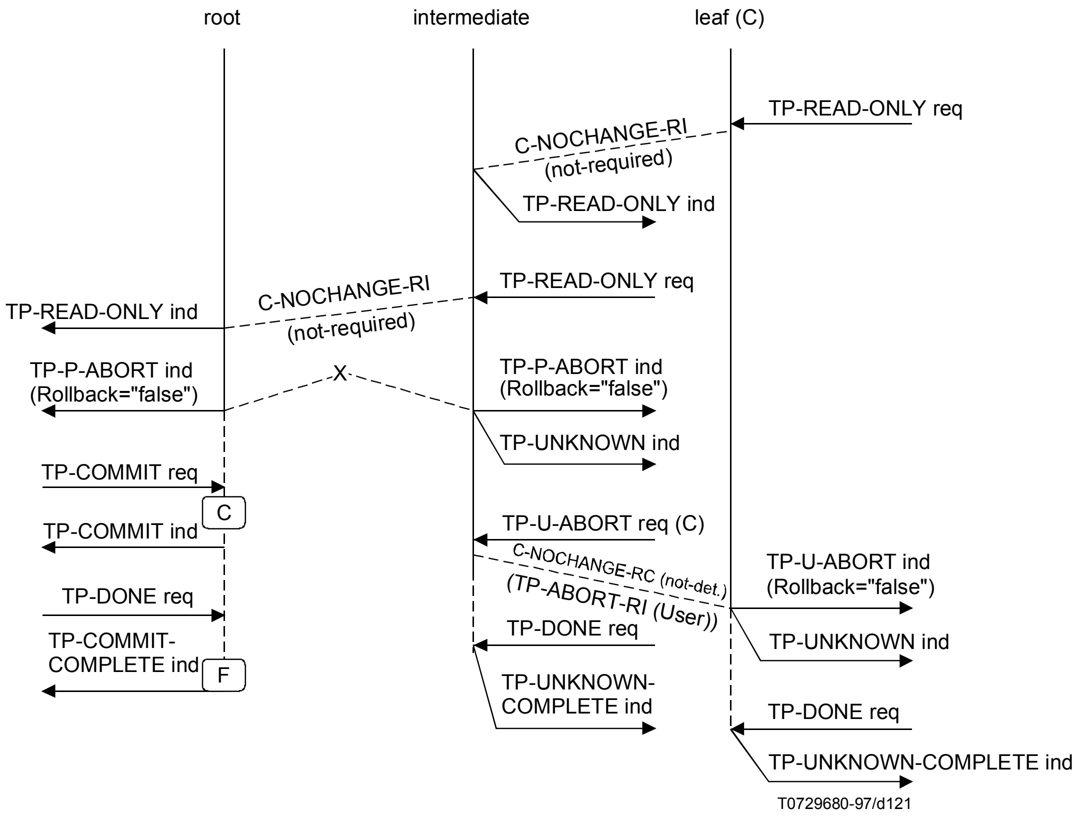


Figure C.115/X.862 – Read-Only Intermediate refuses to become Root Node

C.12 Early-exit scenarios

C.12.1 TP-EARLY-EXIT request with unchained transactions

The scenario of Figure C.116 shows the use of the Early-Exit functional unit with the Commit and Unchained Transactions functional units. At the end of this scenario, the coordination level of the dialogue is none and the subordinate is no longer involved in the transaction. The superior will continue with the transaction.

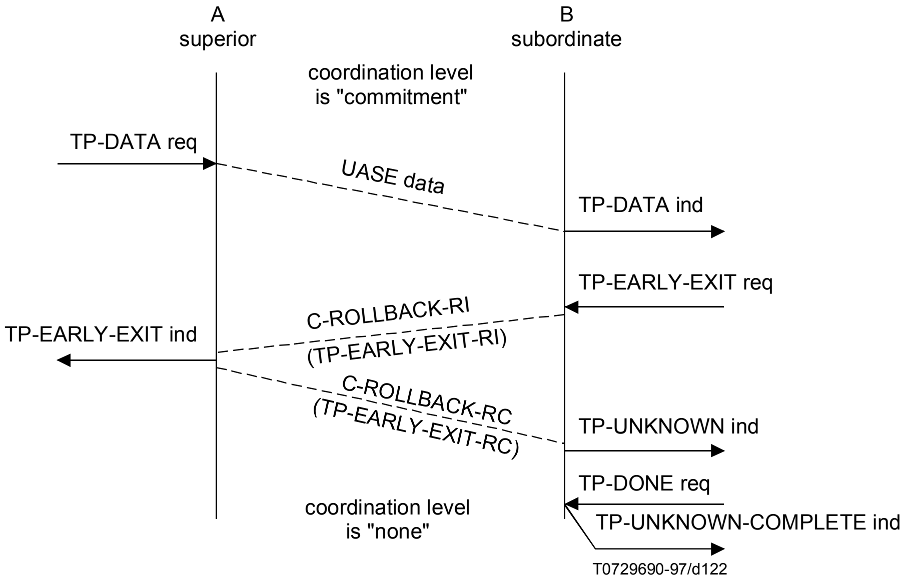


Figure C.116/X.862 – TP-EARLY-EXIT request with unchained transactions

C.12.2 TP-EARLY-EXIT request with chained transactions

The scenario of Figure C.117 shows the use of the Early-Exit functional unit with the Commit and Chained Transactions functional units. On receiving the TP-EARLY-EXIT indication, the superior proceeds to commitment; the dialogue to the subordinate is blocked awaiting the C-BEGIN-RI for the next transaction in the chain.

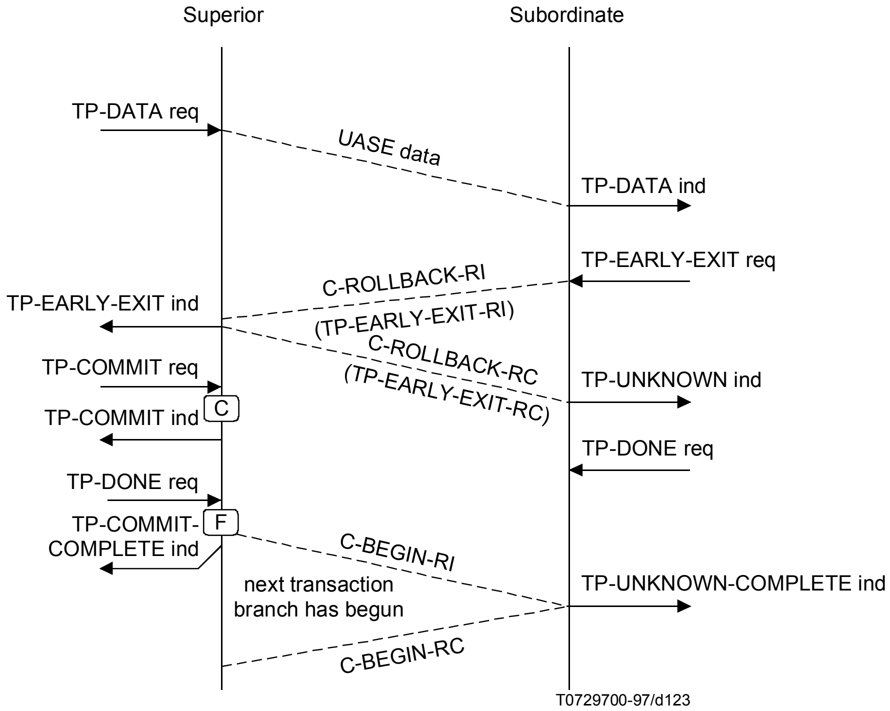


Figure C.117/X.862 – TP-EARLY-EXIT request with chained transactions

C.12.3 TP-EARLY-EXIT request in response to TP-PREPARE request – Unchained transactions

The scenario of Figure C.118 shows the use of the Early-Exit functional unit with the Commit and Unchained Transactions functional units. At the end of this scenario, the coordination level of the dialogue is none and the subordinate is no longer involved in the transaction. The superior will continue with the transaction.

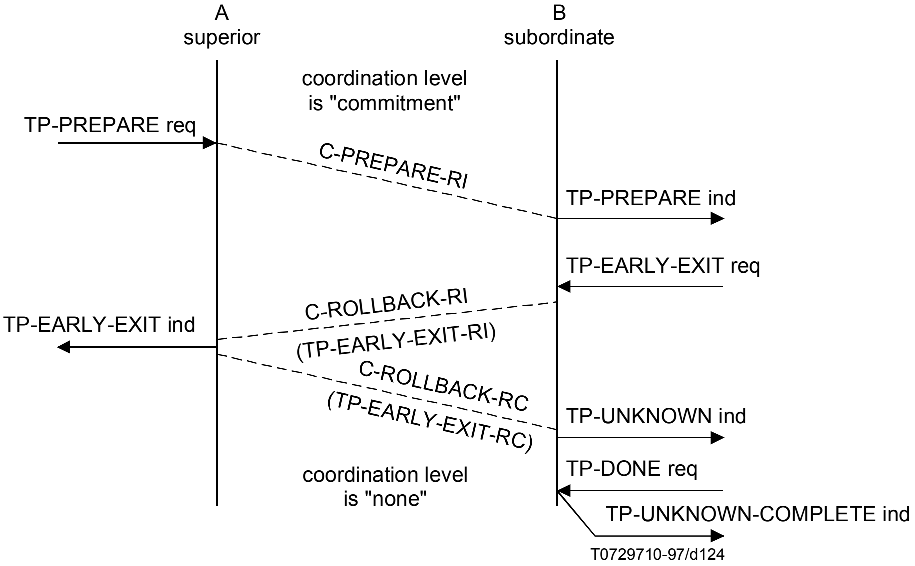


Figure C.118/X.862 – TP-EARLY-EXIT request in response to TP-PREPARE request – Unchained transactions

C.12.4 TP-EARLY-EXIT request collision with TP-PREPARE request – Unchained transactions

The scenario of Figure C.119 shows the use of the Early-Exit functional unit with the Commit and Unchained Transactions functional units. At the end of this scenario, the coordination level of the dialogue is none and the subordinate is no longer involved in the transaction. The superior will continue with the transaction.

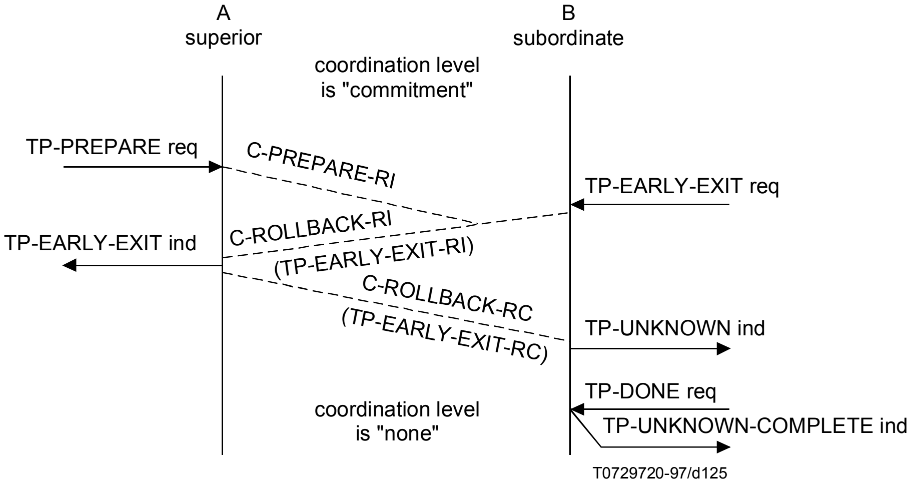


Figure C.119/X.862 – TP-EARLY-EXIT request collision with TP-PREPARE request – Unchained transactions

C.12.5 Repeated use of a dialogue with early-exit and unchained transactions

The scenario of Figure C.120 is that an application passes a request to a subordinate; if on examining the request the subordinate discovers that it can perform no useful processing, it early-exits to free the dialogue. The requestor can then perform further actions on the dialogue, e.g. begin another transaction branch and pass a further request. In the scenario of Figure C.120, the use of Implicit Prepare and Shared Control is assumed in order to simplify the diagram.

A similar scenario would be possible with TP-READ-ONLY request in place of TP-EARLY-EXIT request.

Such effects are not possible with chained transactions as the dialogue becomes blocked waiting for the next transaction in the chain.

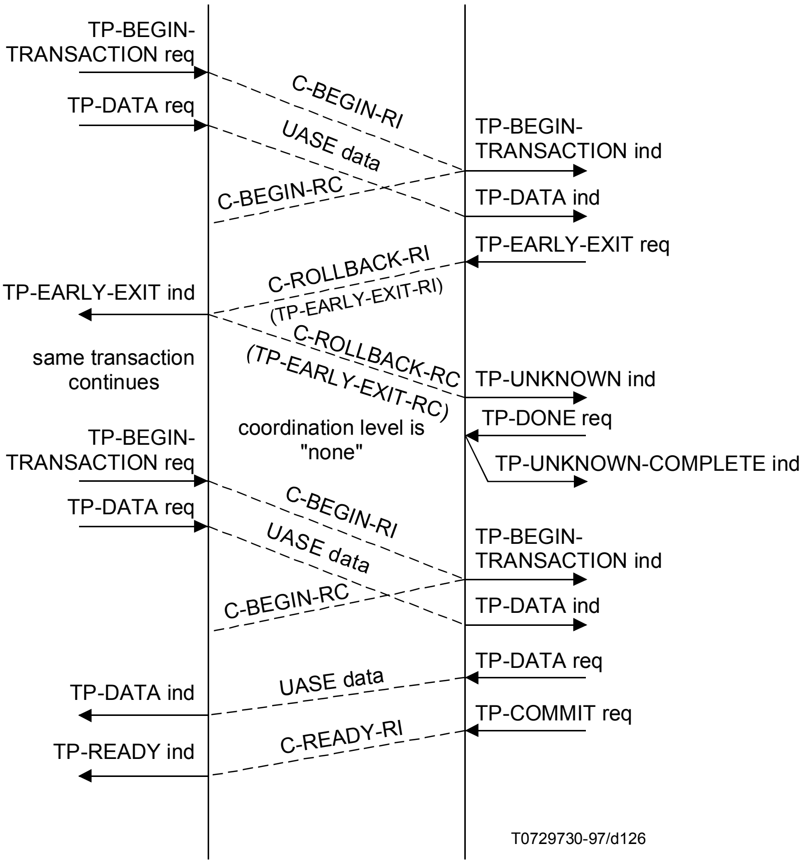


Figure C.120/X.862 – Repeated use of a dialogue with early-exit and unchained transactions

C.12.6 TP-EARLY-EXIT request and TP-COMPLETION-REPORT indication

If the Completion Diagnostics functional unit is selected, then the TP-EARLY-EXIT indication will carry the diagnostic values that were provided on the TP-EARLY-EXIT request. If however, the early exit report arrives during the transaction termination phase at the superior, then this information will be reported by a TP-COMPLETION-REPORT indication as shown in the following scenarios, in each case with the Diagnostic parameter set to "early-exit-transaction-completion-collision". The cases shown are for Unchained Transactions.

C.12.6.1 TP-EARLY-EXIT request – No collision

Firstly the no collision case.

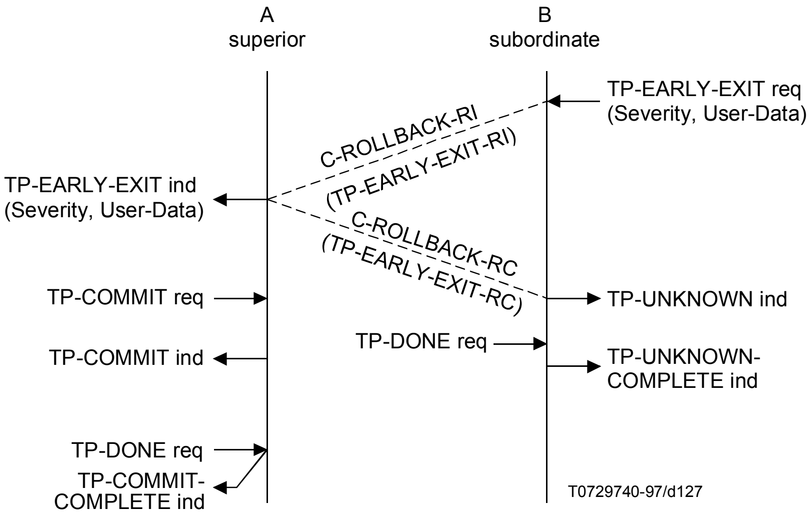


Figure C.121/X.862 – TP-EARLY-EXIT request – No collision

C.12.6.2 Collision with transaction completion

Secondly, a collision between the early exit request and a transaction completion request at the superior. The transaction is rolled back as this is treated as a TP user error; a completion report is issued with the diagnostics supplied with the early exit request. This collision is also detected at the subordinate as the rollback confirmation does not carry an early exit confirmation.

Collision with TP-COMMIT request is shown; similar results would occur if A issued a TP-ONE-PHASE request, or a TP-READ-ONLY request (in which case A would necessarily be an intermediate node).

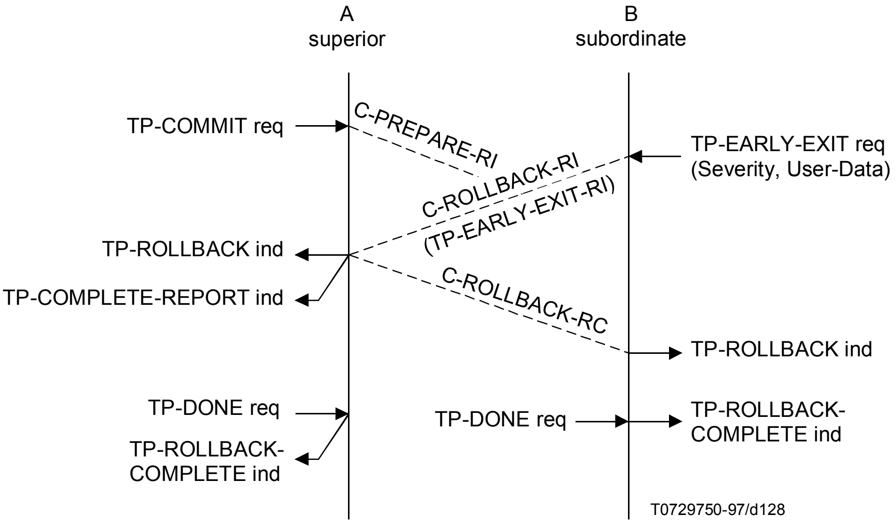


Figure C.122/X.862 – Collision with transaction completion

C.12.6.3 Collision of early exit and rollback (1)

Thirdly a collision between the early exit and a full rollback from the superior but the early exit parameters are delivered with the winning C-ROLLBACK-RI.

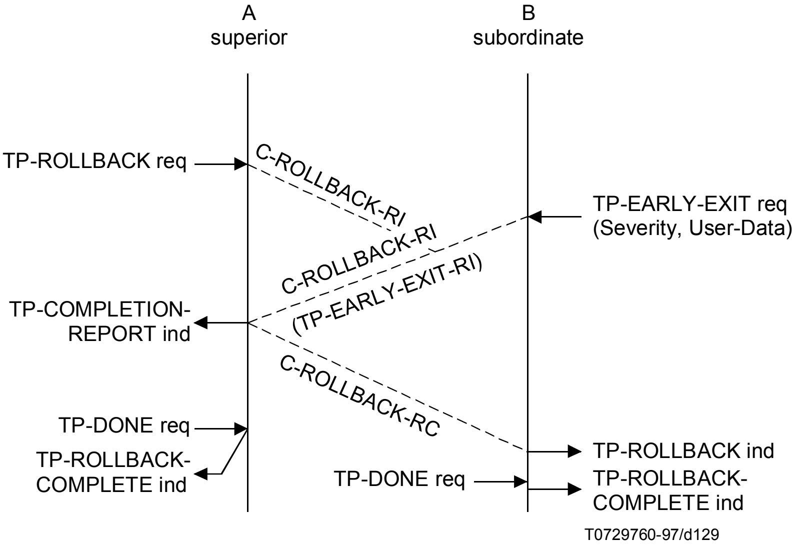


Figure C.123/X.862 – Collision of early exit and rollback (1)

C.12.6.4 Collision of early exit and rollback (2)

Fourthly, a similar collision between the early exit and a full rollback from the superior; the early exit parameters are repeated by a TP-REPORT-RI on the rollback confirmation.

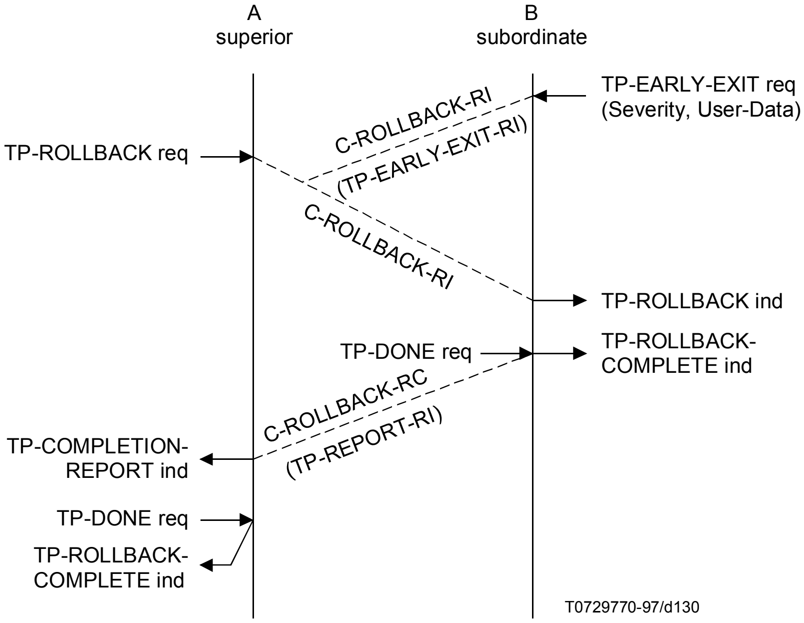


Figure C.124/X.862 – Collision of early exit and rollback (2)

C.12.7 Early exit with one read-only subordinate

Fast shutdown of a subordinate tree with early exit and read-only and unchained transactions.

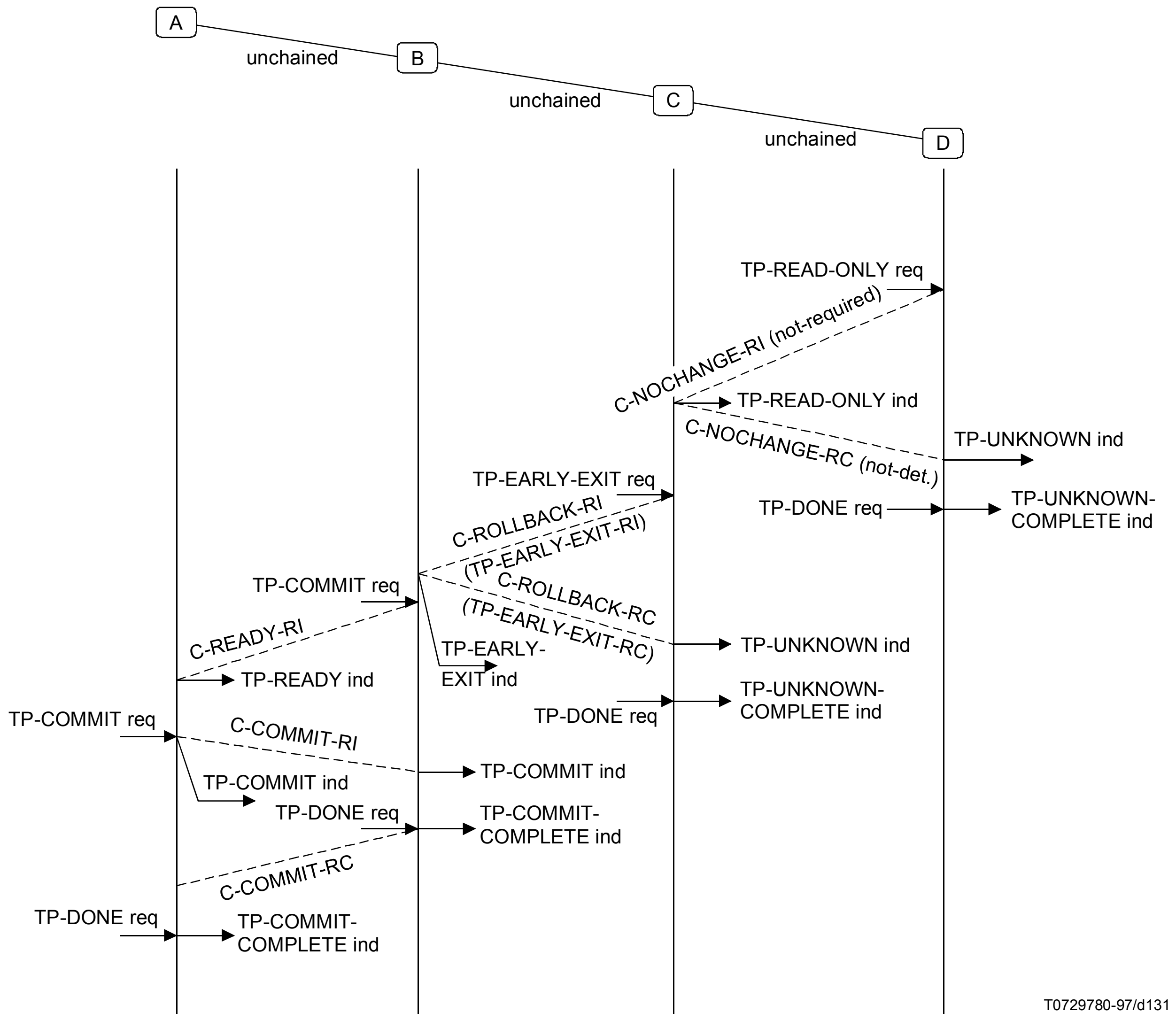


Figure C.125/X.862 – Early exit with one read-only subordinate

C.12.8 Early exit with one read-only subordinate

Fast shutdown of a subordinate tree with early exit and read-only and chained transactions.

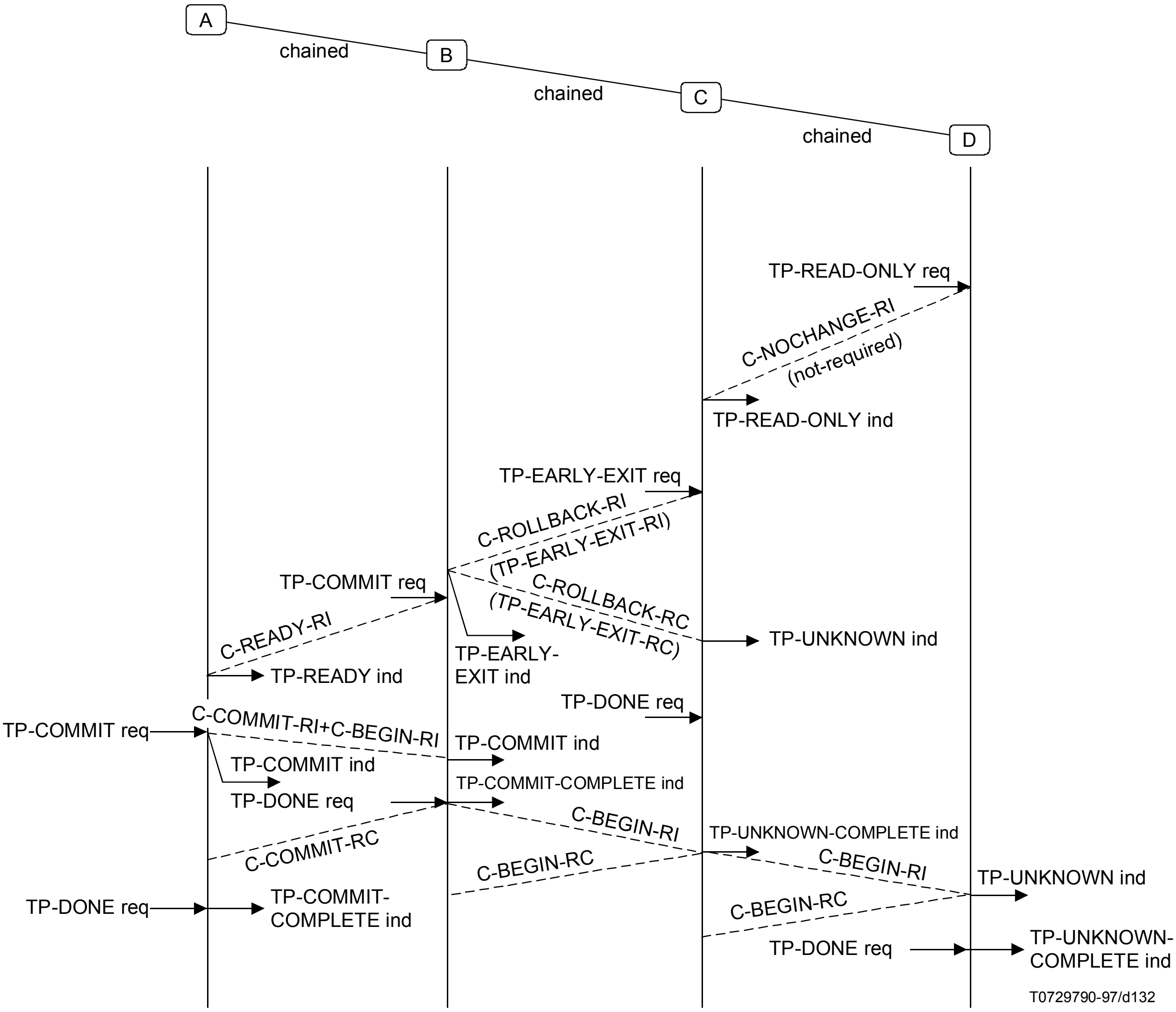


Figure C.126/X.862 – Early exit with one read-only subordinate

C.12.9 Early exit with one read-only subordinate – TP-P-ABORT indication after requesting early exit

After TP-P-ABORT indication, node C is root of the transaction tree and continues with 'undetermined'.

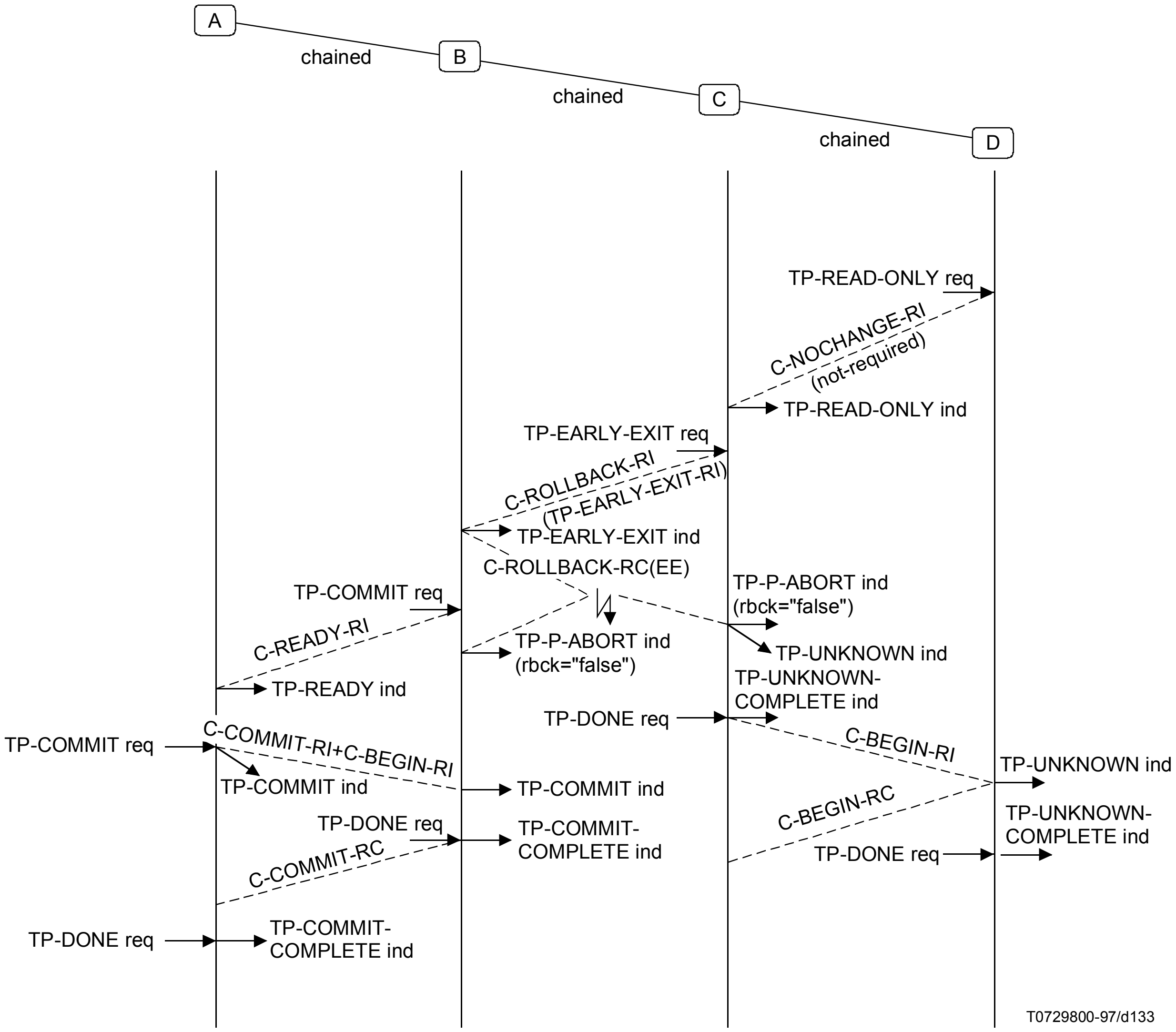


Figure C.127/X.862 – Early exit with one read-only subordinate – TP-P-ABORT indication after requesting early exit

C.12.10 Early exit with one read-only subordinate – TP-P-ABORT indication after requesting early exit

After TP-P-ABORT indication, node C is root of the transaction tree and continues with 'undetermined'. TPPM at node B initiates rollback.

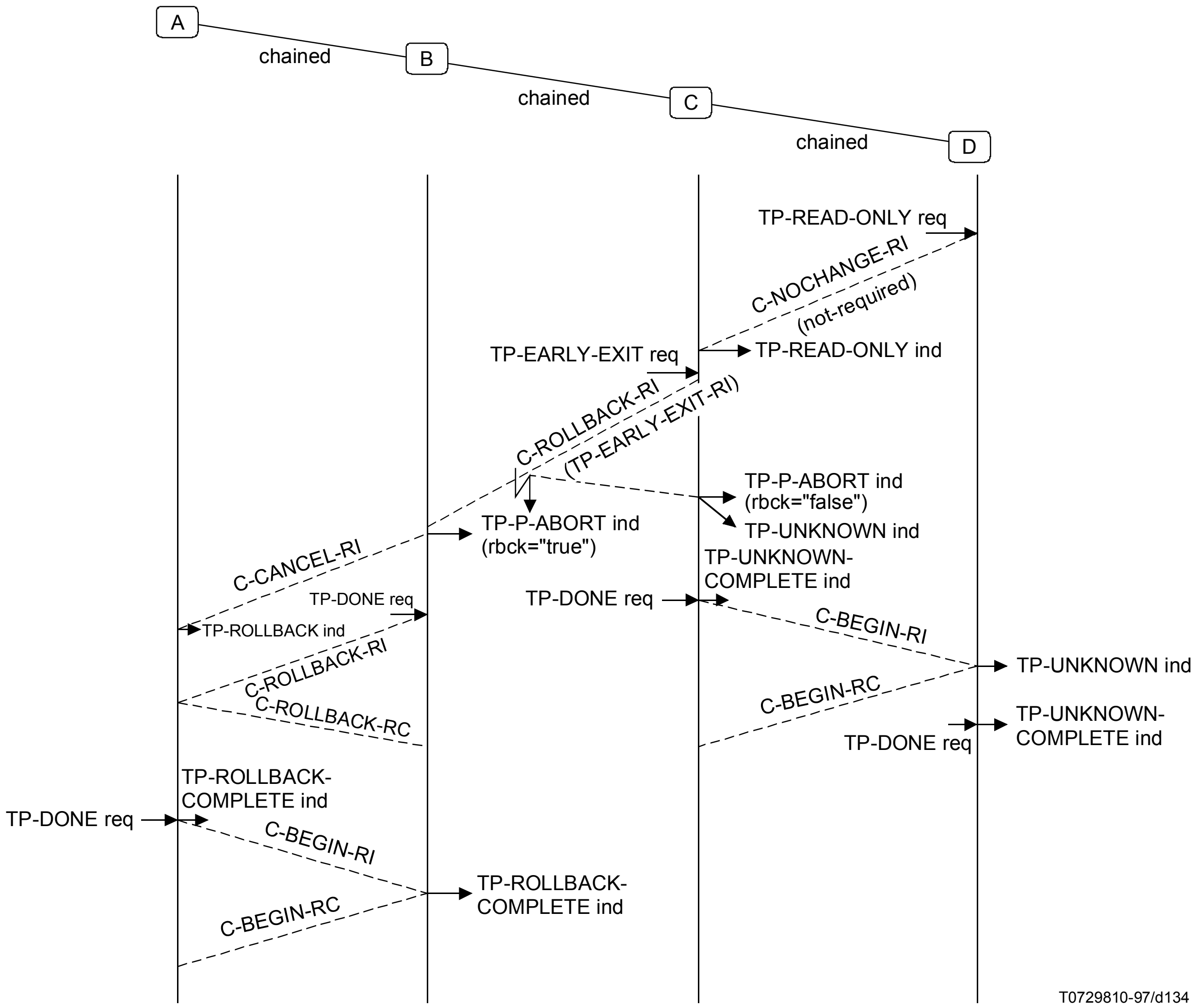


Figure C.128/X.862 – Early exit with one read-only subordinate – TP-P-ABORT indication after requesting early exit

C.12.11 Early exit with one read-only subordinate – TP-P-ABORT indication on a subordinate dialogue after requesting early exit

Node C issues a TP-EARLY-EXIT request; however this arrives late at B and so a full rollback of the transaction is initiated from node B. As a separate event, the dialogue C-D aborts; after receiving TP-P-ABORT indication, failure related actions are allowed at node C.

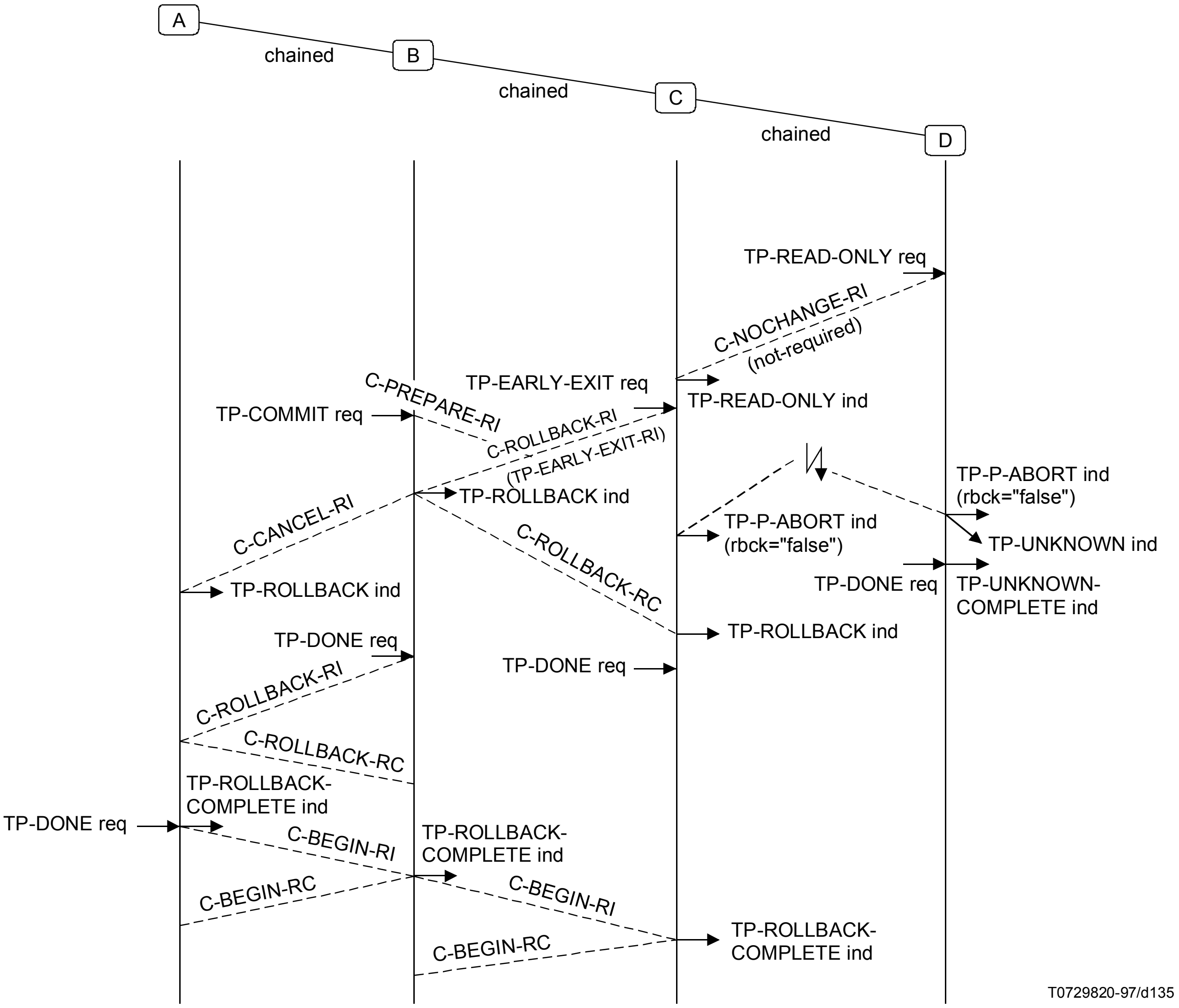


Figure C.129/X.862 – Early exit with one read-only subordinate – TP-P-ABORT indication on a subordinate dialogue after requesting early exit

C.12.12 Early exit with one read-only subordinate – TP-P-ABORT indication after requesting early exit

After TP-P-ABORT indication, node C is root of the transaction tree and continues with 'undetermined'. TP-P-ABORT indication in active phase of node B does not cause rollback after TP-EARLY-EXIT indication. The chained transaction case is similar. (Resolution of RB13 needed.)

C.12.13 Early exit in the termination phase

Early exit in termination causes rollback, read-only in termination phase does not cause rollback.

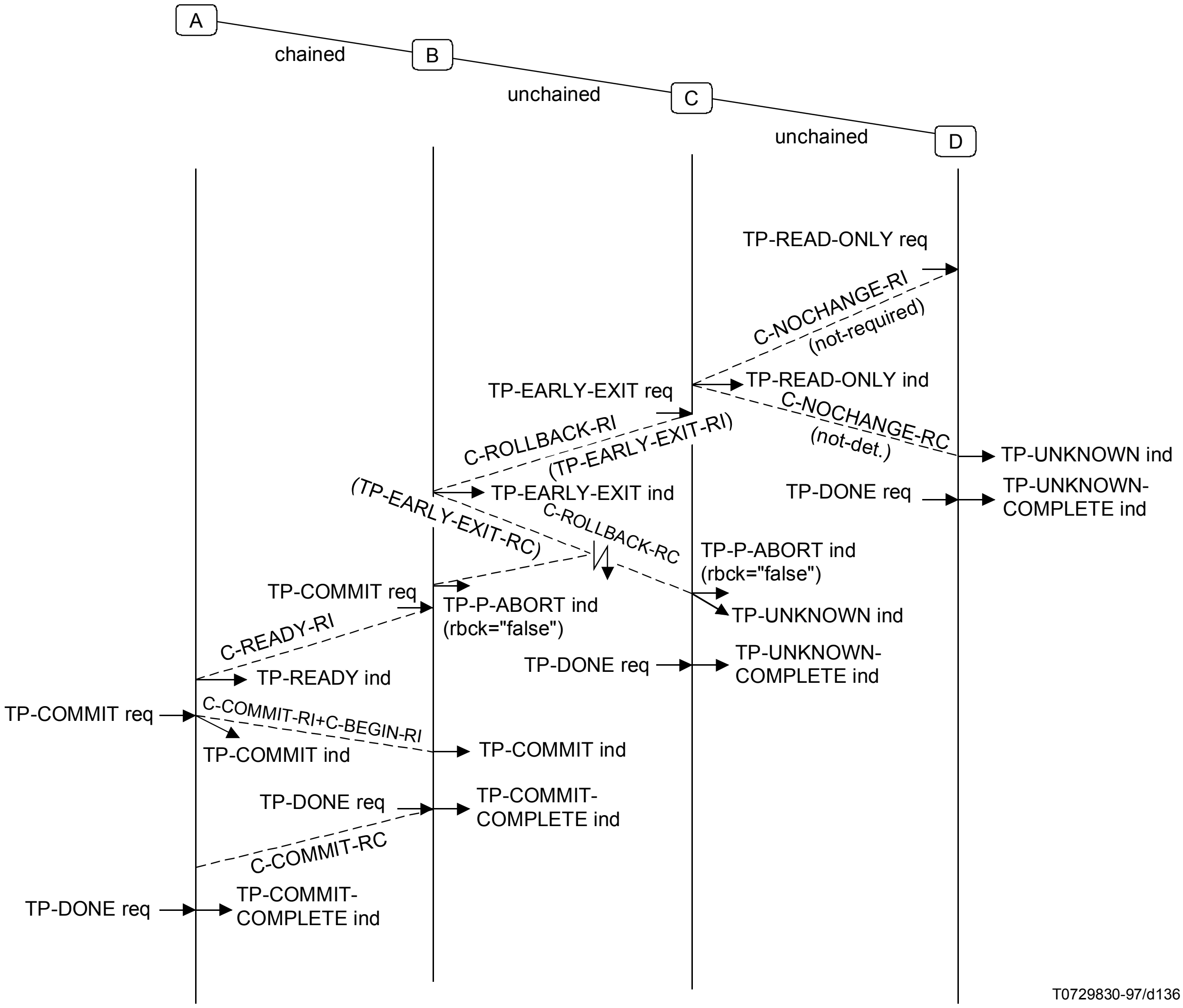


Figure C.130/X.862 – Early exit with one read-only subordinate – TP-P-ABORT indication after requesting early exit

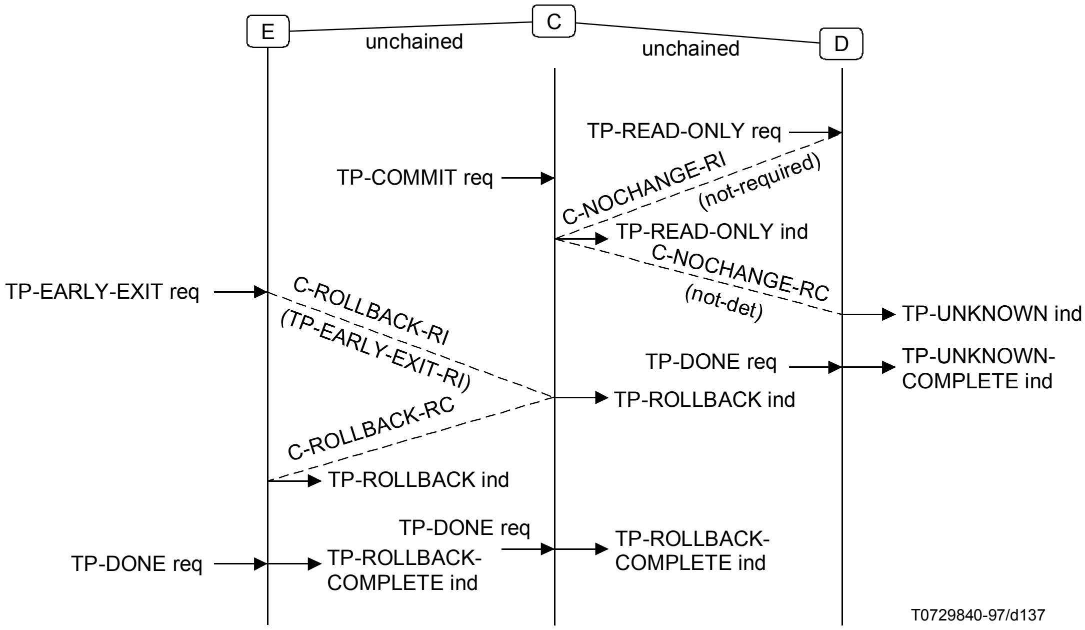


Figure C.131/X.862 – Early exit in the termination phase

C.12.14 Early exit with one read-only subordinate and defer-end-dialogue

After early exit agreement at node B defer-actions for the dialogue B-C are cancelled. Defer actions for the dialogue C-D are performed. The result is unknown at node C and D.

C-NOCHANGE-RI (not-required) is sent from D to C as the full result is not required at D; D simply needs to know whether the transaction has rolled back and so whether the deferred action should be implemented or cancelled. C delays the C-NOCHANGE-RC until it knows that the transaction did not roll back which is indicated in this case by the arrival of the TP-EARLY-EXIT-RC from node B.

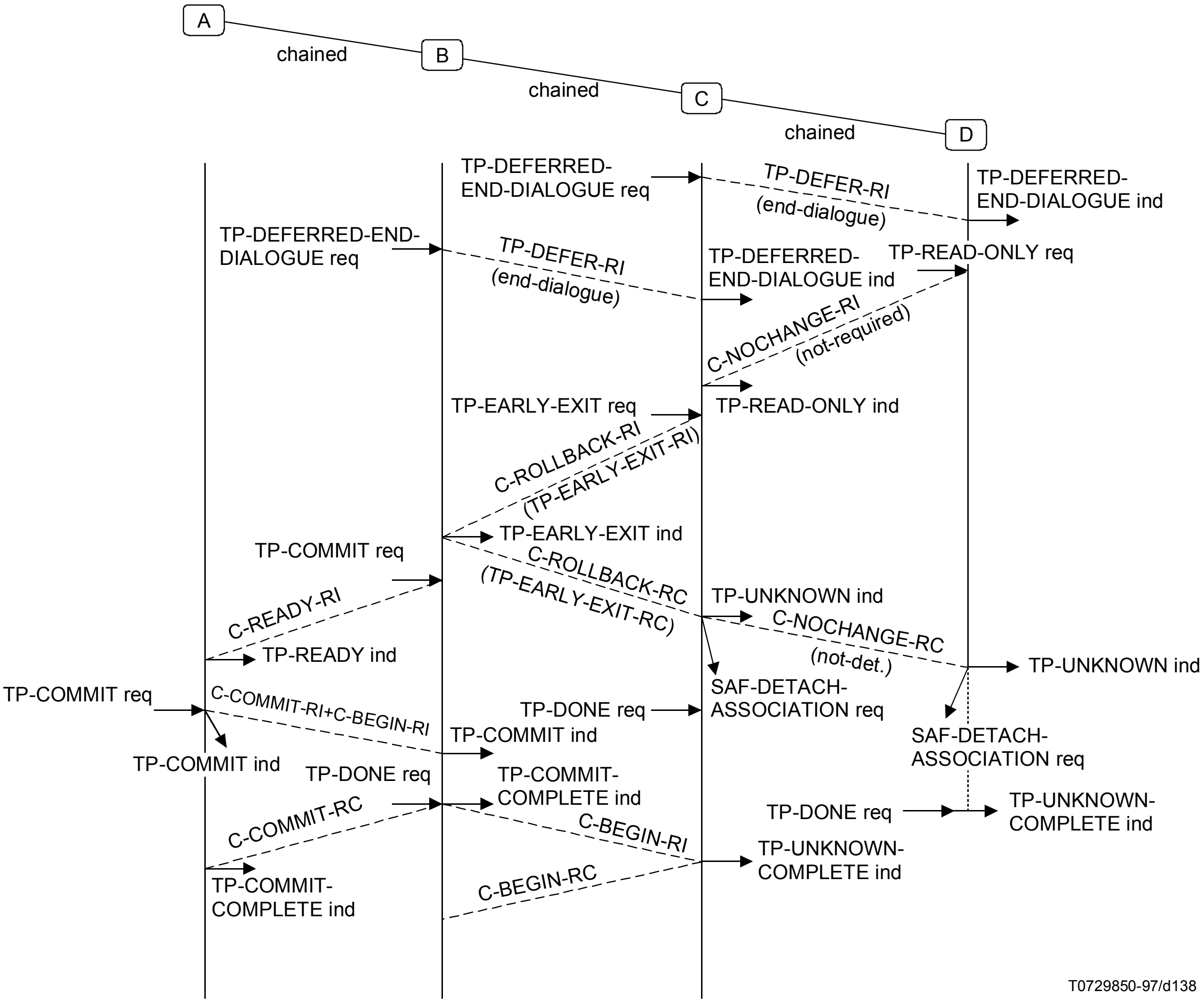


Figure C.132/X.862 – Early exit with one read-only subordinate and defer-end-dialogue

C.12.15 Early exit with one read-only subordinate and defer-end-dialogue in unchained

Early-exit on the B-C dialogue cancels defer-actions for the dialogue. Defer actions for the read-only dialogue C-D are performed. The result is unknown at node C and D. (This is an allowed, but questionable use of Early-exit.)

C-NOCHANGE-RI (not-required) is sent from D to C as the full result is not required at D; D simply needs to know whether the transaction has rolled back and so whether the deferred action should be implemented or cancelled. C delays the C-NOCHANGE-RC until it knows that the transaction did not roll back which is indicated in this case by the arrival of the TP-EARLY-EXIT-RC from node B.

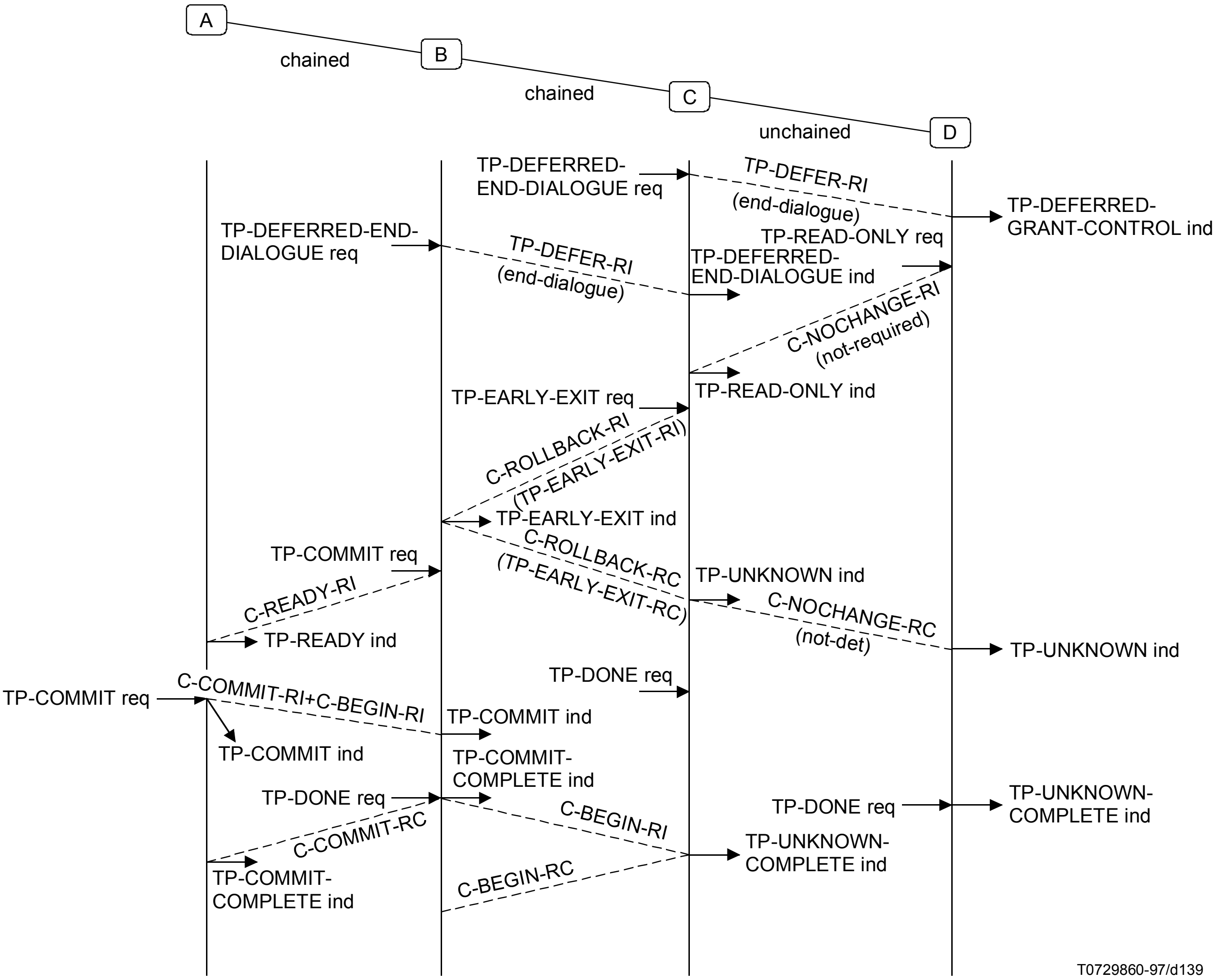


Figure C.133/X.862 – Early exit with one read-only subordinate and defer-end-dialogue in unchained

C.12.16 Early exit above a read-only dialogue with defer-grant-control in unchained – Subordinate aborts the dialogue

Similar, but D wants to abort (failure-related actions permitted by some other failure, not shown). The timing of the abort request at D (before or after the TP-UNKNOWN indication) and the first DONE at C will have minor impact on this scenario (if after the abort, there is only one TP-DONE request at C).

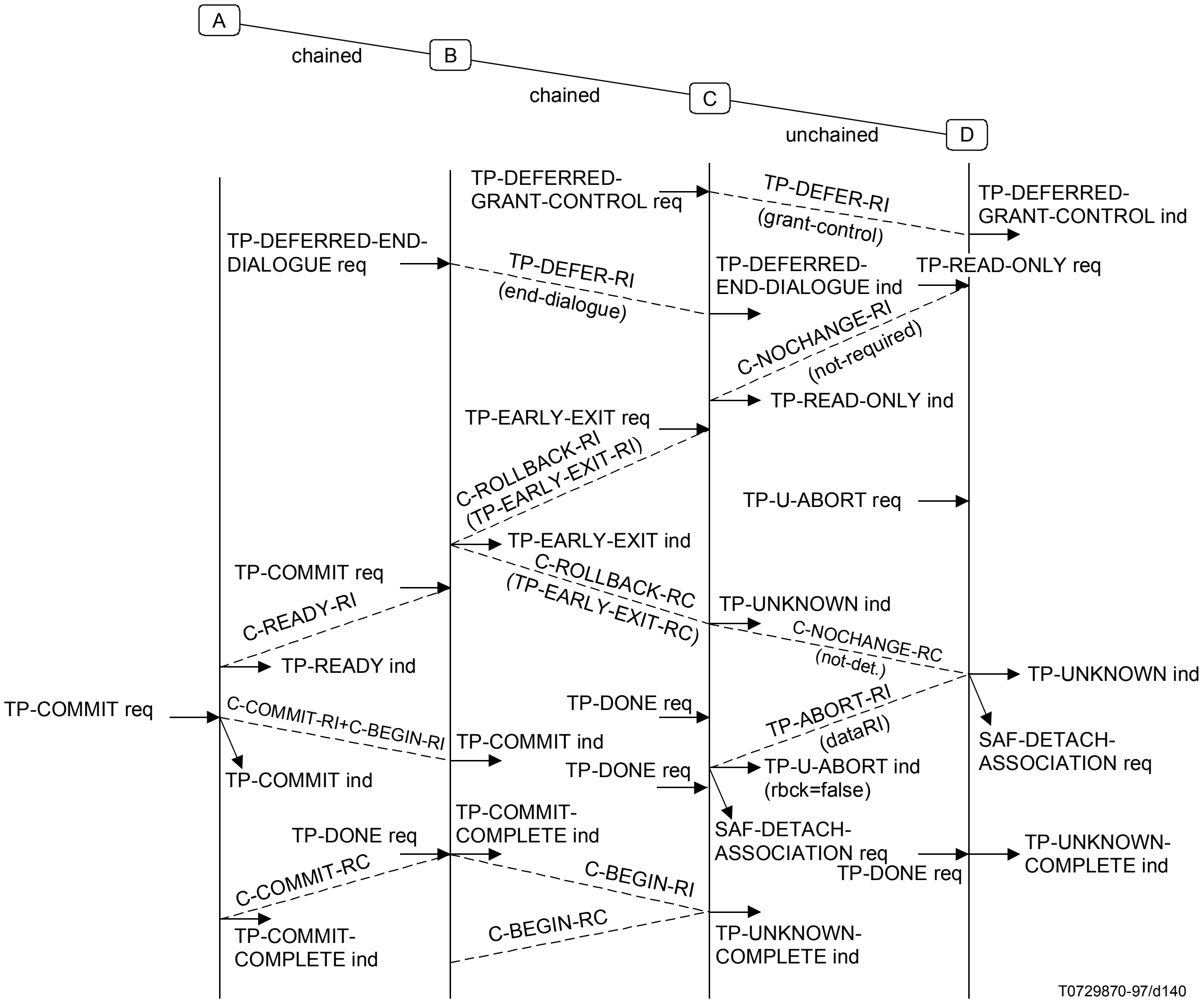


Figure C.134/X.862 – Early exit above a read-only dialogue with defer-grant-control in unchained – Subordinate aborts the dialogue

C.12.17 Early exit and collision with defer

Node B issues a TP-DEFERRED-END-DIALOGUE request but this collides with a TP-EARLY-EXIT request from node C; the defer is purged by the C-ROLLBACK-RI and so is not signalled at node C. The TP-EARLY-EXIT indication at node B cancels the defer action.

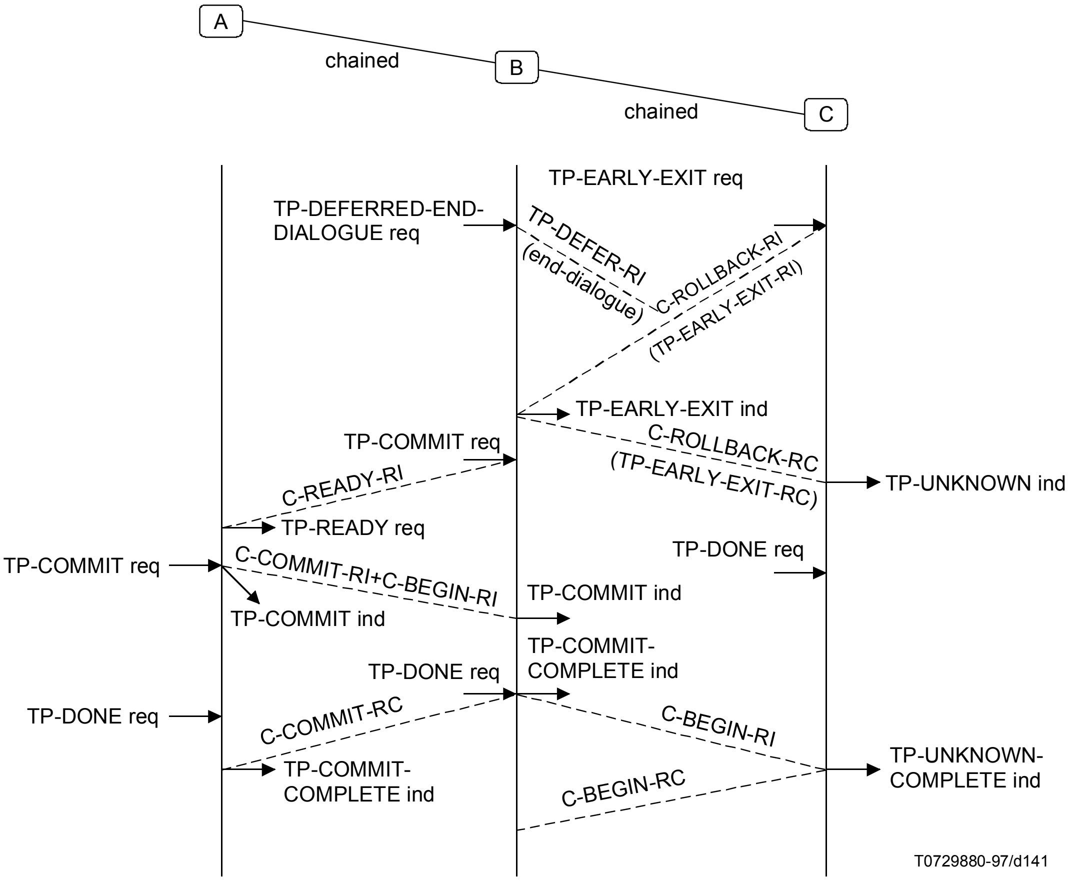


Figure C.135/X.862 – Early exit and collision with defer

C.13 Static one-phase commitment scenarios

Prior to the issuing of a TP-ONE-PHASE request, scenarios with static one-phase commit are identical to those with (static) two phase commit; for example, the superior or subordinate may issue a TP-ROLLBACK request and the normal rollback procedures apply; also dialogue abort scenarios are the same before TP-ONE-PHASE request is issued.

C.13.1 TP-ONE-PHASE request with unchained transactions

The scenario of Figure C.136 shows the use of the static One-phase procedures with Unchained Transactions, i.e. the One-phase functional unit is selected but the Commit functional unit is not selected. The Heuristic Report Suppression functional unit is selected. The outcome is commitment.

From an OSI TP perspective, there is no requirement for a commit log at the subordinate because there is no obligation to perform recovery; however there must be a local equivalent of a commit log to ensure that the ACID properties are not violated at the subordinate.

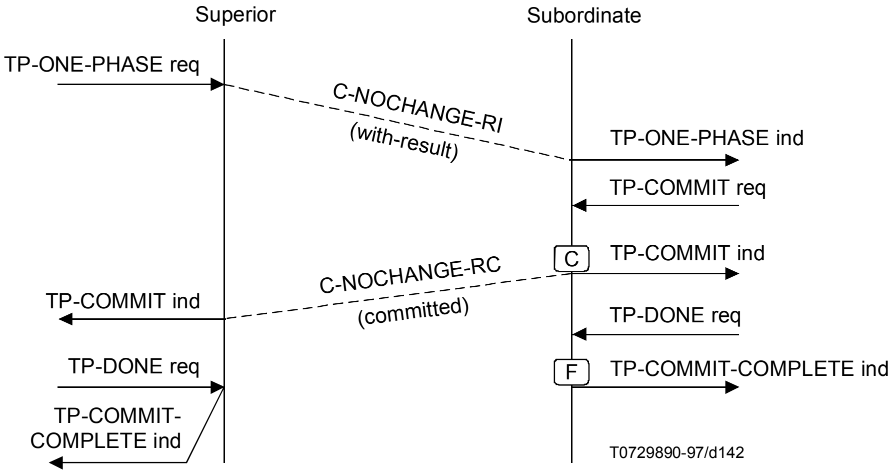


Figure C.136/X.862 – TP-ONE-PHASE request with unchained transactions

C.13.2 TP-ONE-PHASE request with chained transactions

The scenario of Figure C.137 shows the use of the static One-phase procedures with Chained Transactions, i.e. the One-phase functional unit is selected but the Commit functional unit is not selected. The Heuristic Report Suppression functional unit is selected. The outcome is commitment.

From an OSI TP perspective, there is no requirement for a commit log at the subordinate because there is no obligation to perform recovery; however there must be a local equivalent of a commit log to ensure that the ACID properties are not violated at the subordinate.

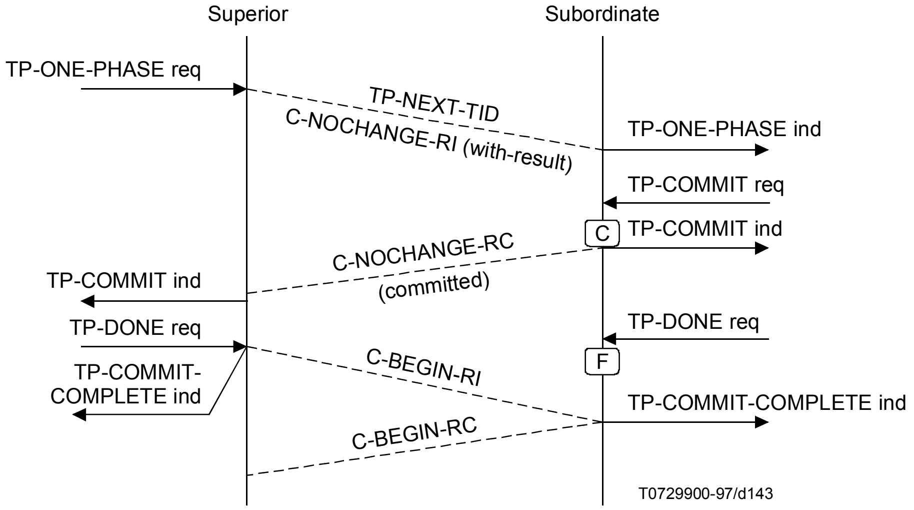


Figure C.137/X.862 – TP-ONE-PHASE request with chained transactions

C.13.3 TP-ONE-PHASE request with unchained transactions – Rollback case

The scenario of Figure C.138 shows the use of the static One-phase procedures with Unchained Transactions, i.e. the One-phase functional unit is selected but the Commit functional unit is not selected. The subordinate rolls back; the normal rollback protocol flows on the branch.

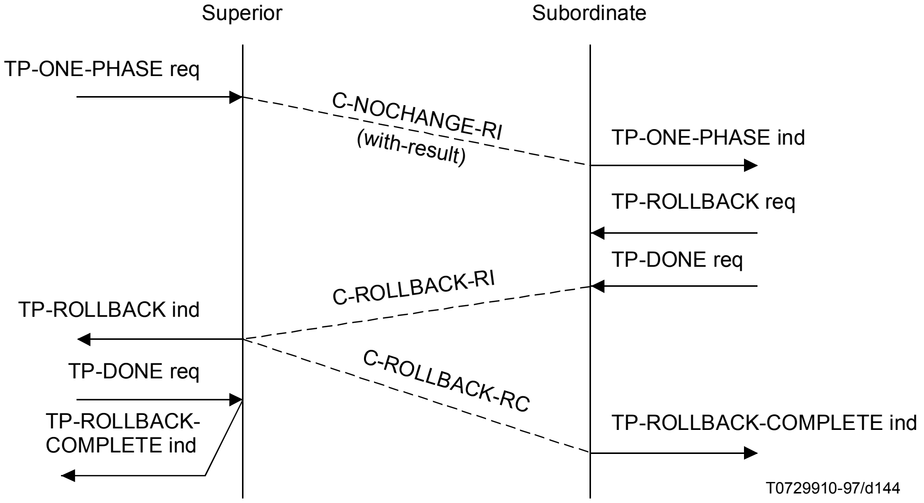


Figure C.138/X.862 – TP-ONE-PHASE request with unchained transactions – Rollback case

C.13.4 TP-ONE-PHASE request with unchained transactions – Dialogue aborts late

The scenario of Figure C.139 shows the use of the static One-phase procedures with Unchained Transactions, i.e. the One-phase functional unit is selected but the Commit functional unit is not selected. The dialogue is aborted after the subordinate commits; the result is not reported to the superior.

From an OSI TP perspective, there is no requirement for a commit log at the subordinate because there is no obligation to perform recovery; however there must be a local equivalent of a commit log to ensure that the ACID properties are not violated at the subordinate.

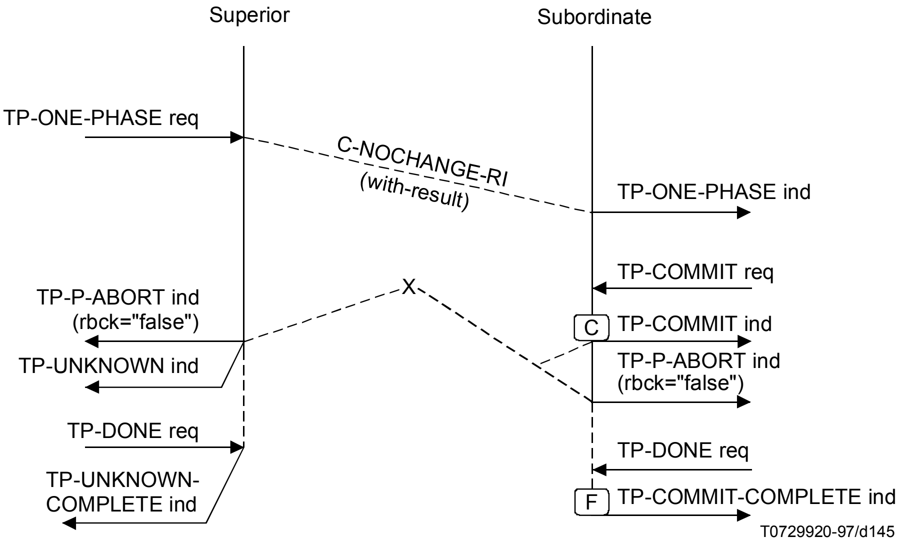


Figure C.139/X.862 – TP-ONE-PHASE request with unchained transactions – Dialogue aborts late

C.13.5 TP-ONE-PHASE request with unchained transactions – Dialogue aborts early

The scenario of Figure C.140 shows the use of the static One-phase procedures with Unchained Transactions, i.e. the One-phase functional unit is selected but the Commit functional unit is not selected. The dialogue is aborted before the subordinate has issued a transaction completion request; however the transaction can continue as the superior has signalled that it has no bound data and so atomicity is not prejudiced; the result is not reported to the superior.

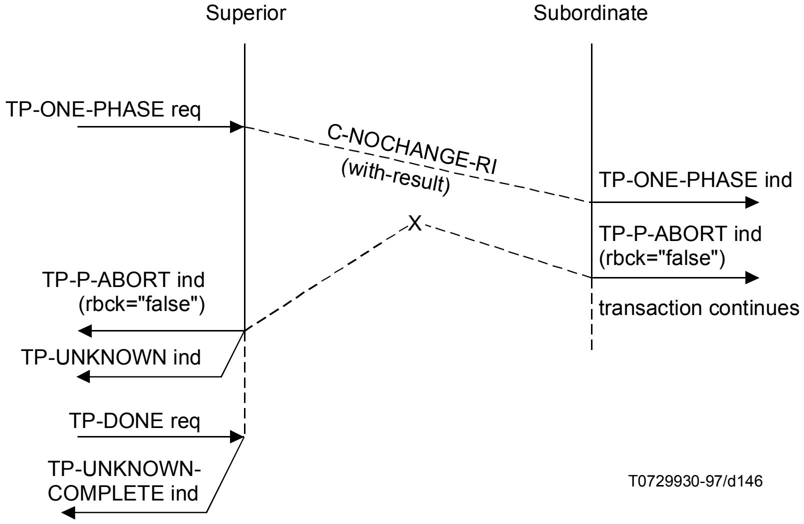


Figure C.140/X.862 – TP-ONE-PHASE request with unchained transactions – Dialogue aborts early

C.13.6 Static one-phase above two-phase – Unchained transactions

The scenario of Figure C.141 shows a scenario, with unchained transactions, in which root A uses one-phase commitment, but B and C use two-phase. The heuristic status is reported to A.

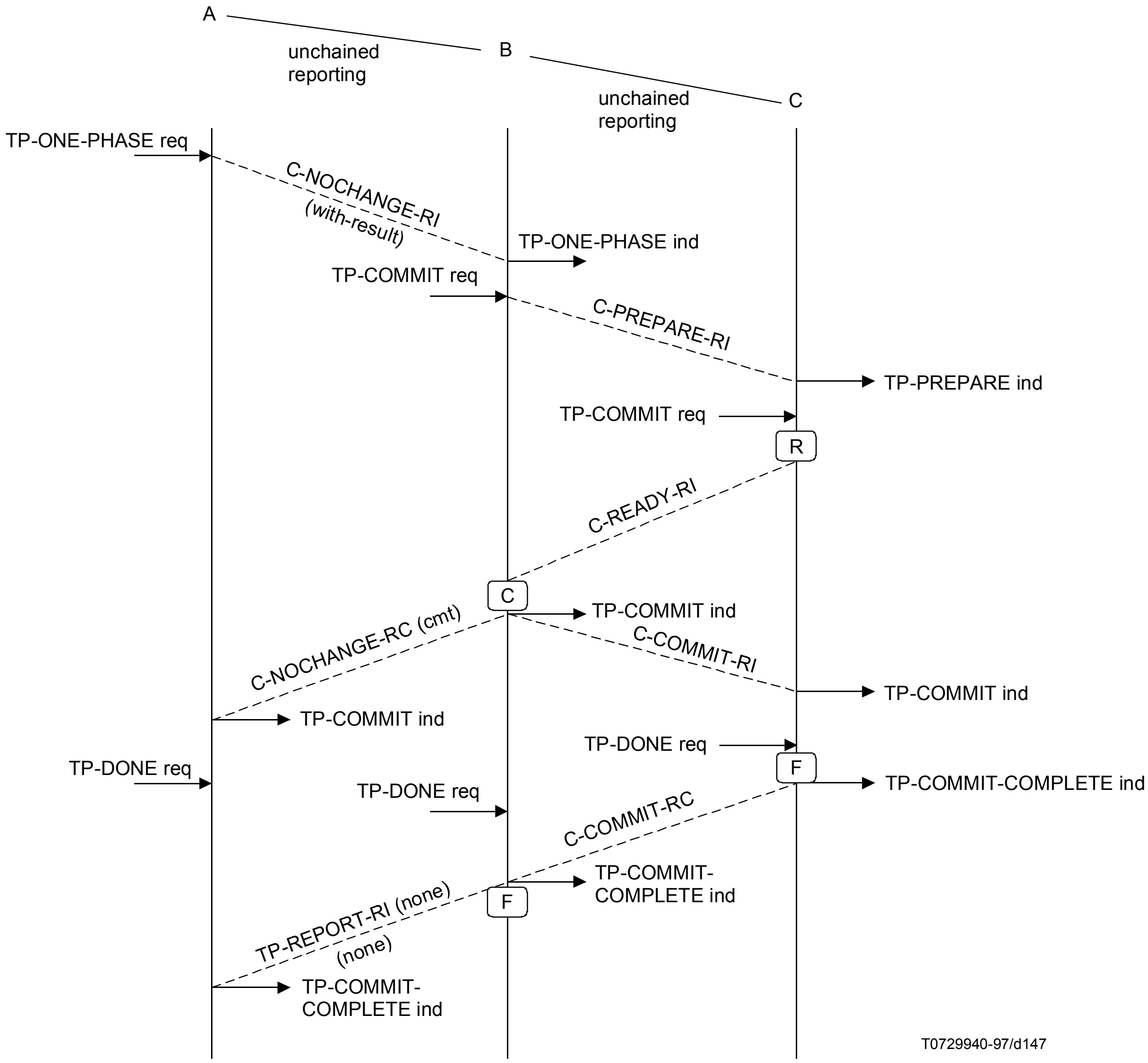


Figure C.141/X.862 – Static one-phase above two-phase – Unchained transactions

C.13.7 Static one-phase above two-phase – Chained transactions

The scenario of Figure C.142 shows a scenario, with chained transactions, in which root A uses one-phase commitment, but B and C use two-phase. The next-transaction identifiers are passed down with the one-phase request, and B uses these to send C-BEGIN-RI to C as part of the two-phase commit sequence before receiving C-BEGIN-RI from A for the same transaction. The heuristic status is reported to A.

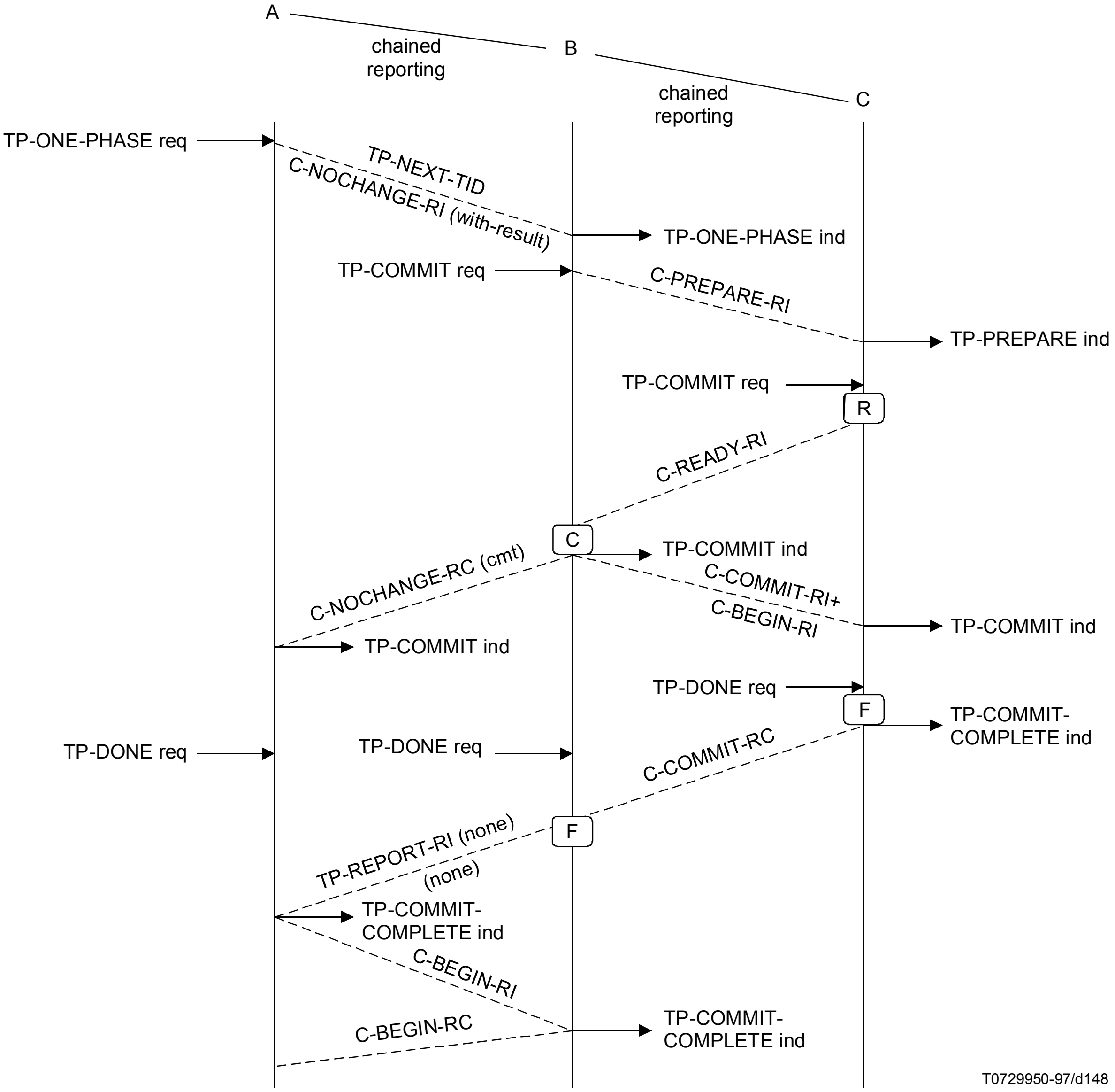


Figure C.142/X.862 – Static one-phase above two-phase – Chained transactions

C.13.8 Static one-phase above two-phase – Chained transactions – Leaf rolls back

The scenario of Figure C.143 shows a scenario, with chained transactions, in which root A uses one-phase commitment, but B requests two-phase commit, but C requests rollback. The next-transaction identifiers are passed down with the one-phase request but are not used. The heuristic status is reported to A.

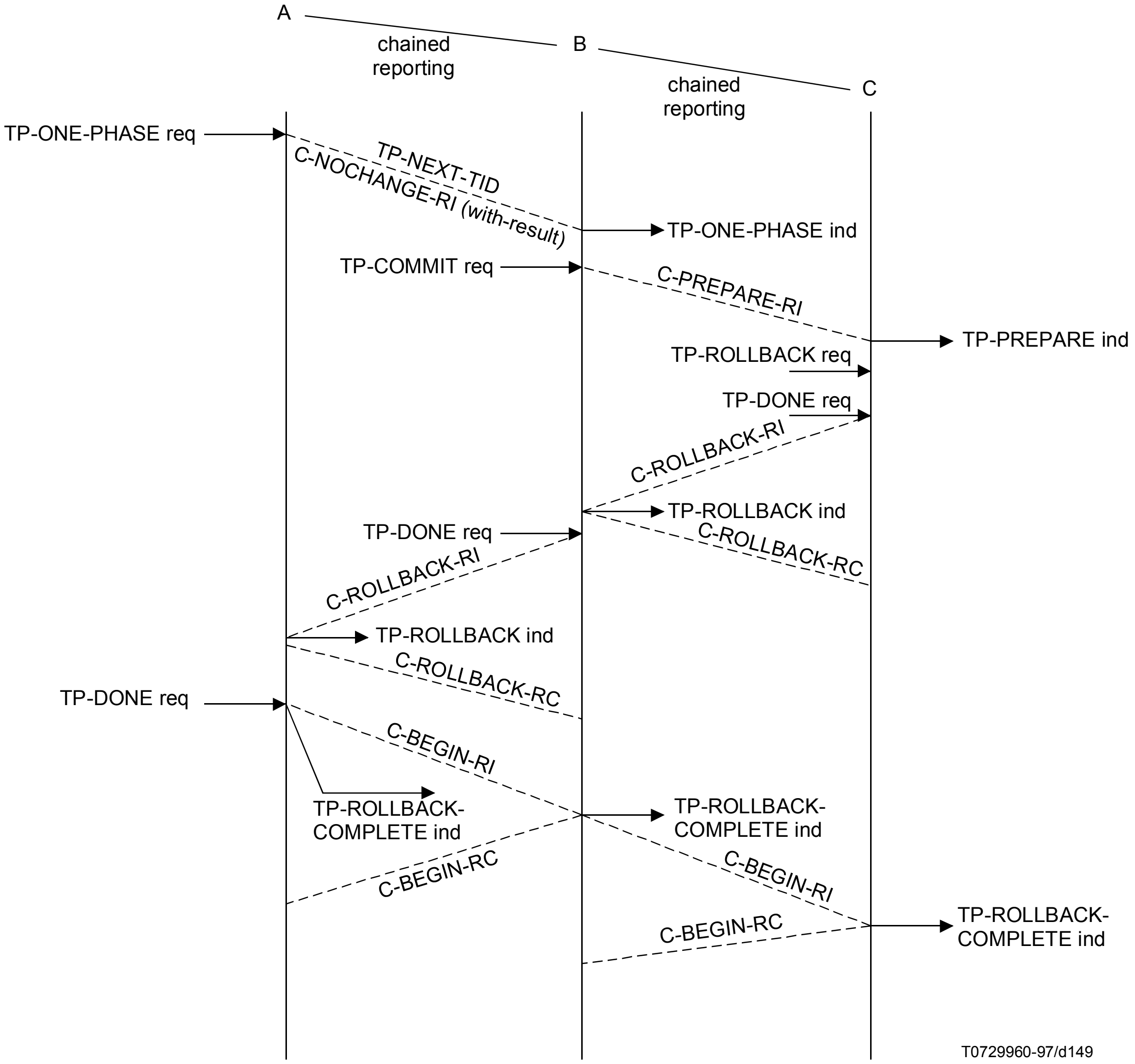


Figure C.143/X.862 – Static one-phase above two-phase – Chained transactions – Leaf rolls back

C.13.9 Static one-phase above two-phase – Chained transactions – No reporting

The next-transaction identifiers are passed down with the one-phase request. Node B issues C-COMMIT+C-BEGIN request after making the commit decision. C-BEGIN indication is awaited on the dialogue with the superior. The Heuristic Report Suppression functional unit is selected on the dialogue A-B. The Implicit Prepare functional unit is selected on dialogue B-C.

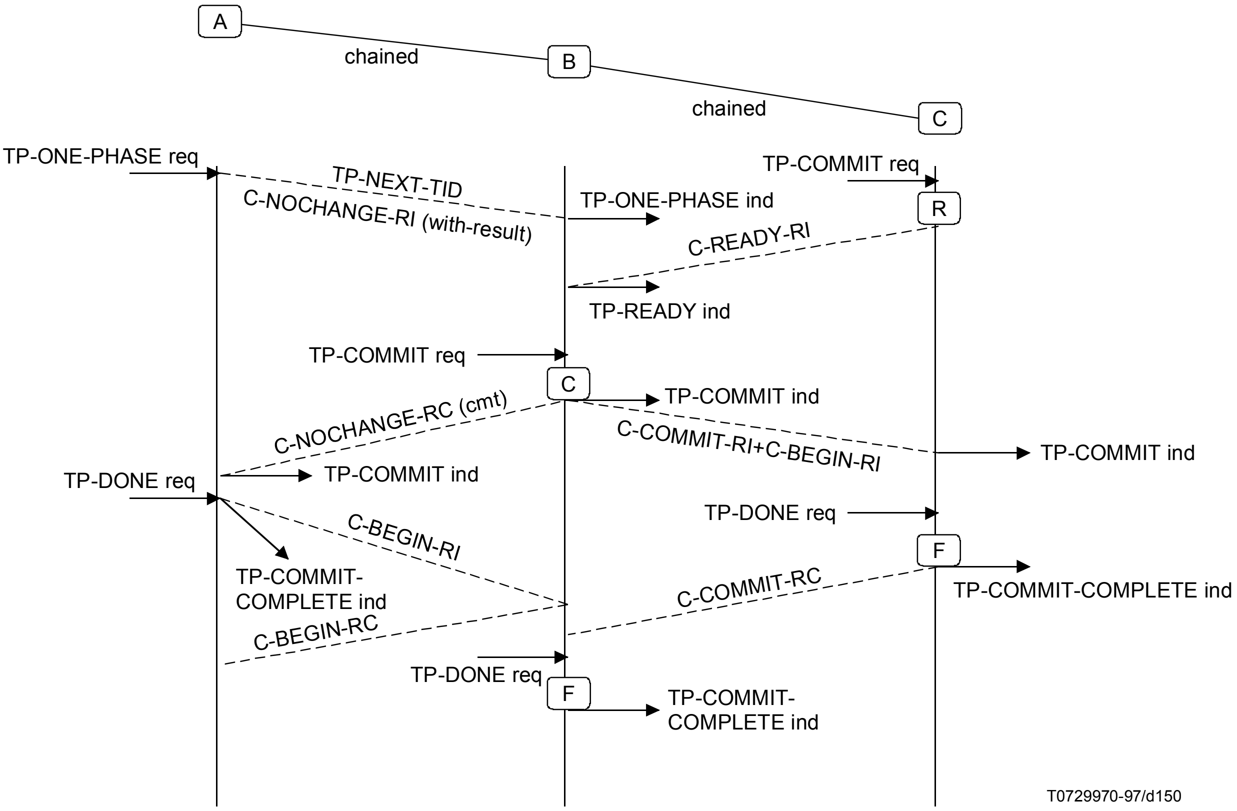


Figure C.144/X.862 – Static one-phase above two-phase – Chained transactions – No reporting

C.13.10 Static one-phase and read-only – Unchained

Figure C.145 shows a "do-nothing" transaction, in which A issues TP-ONE-PHASE request and B and C issue TP-READ-ONLY request; functional units selected:

- Dialogue A-B: One-phase, Unchained Transactions, Read-Only.
- Dialogue B-C: Commit, Unchained Transactions, Read-Only.

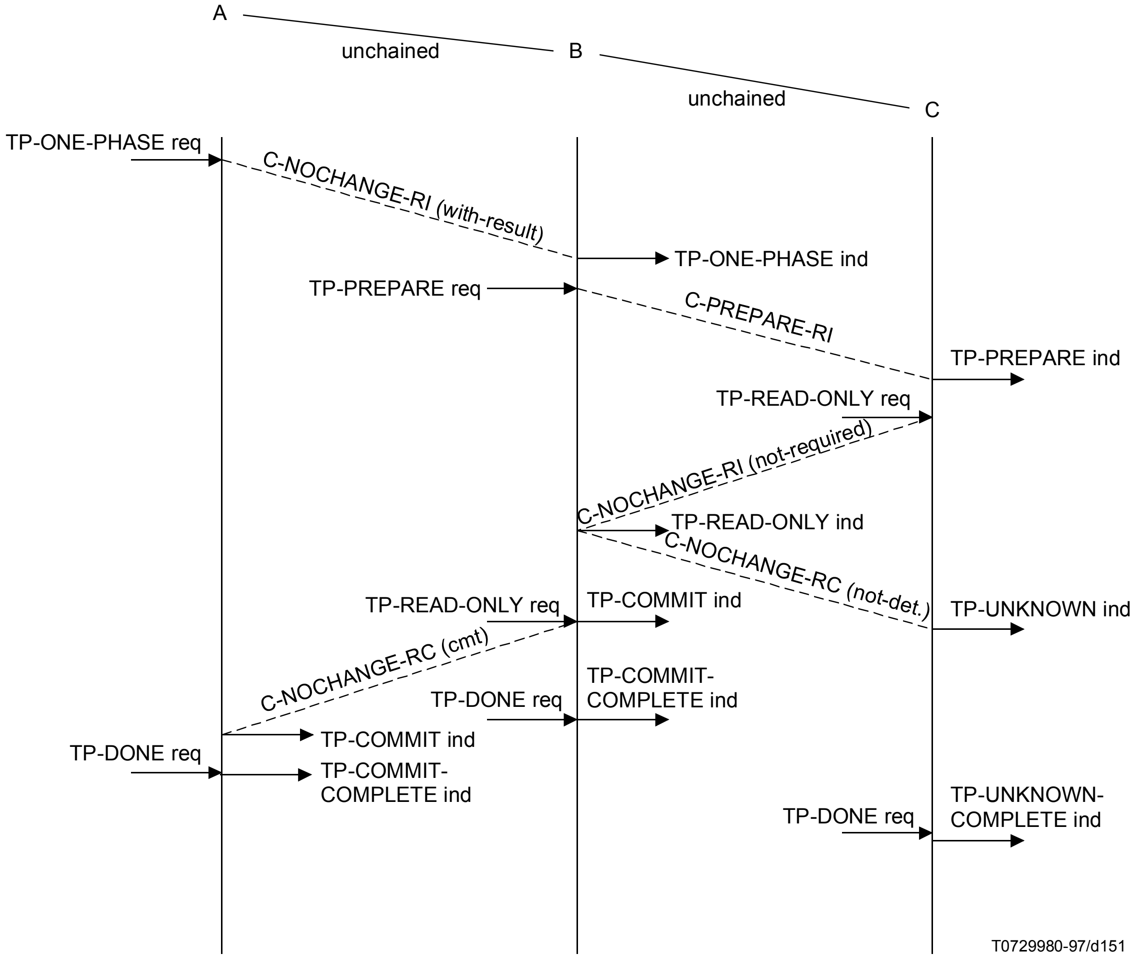


Figure C.145/X.862 – Static one-phase and read-only – Unchained

C.13.11 Static one-phase and read-only – Chained

Figure C.146 shows a "do-nothing" transaction, in which A issues TP-ONE-PHASE request and B and C issue TP-READ-ONLY request; functional units selected:

– Dialogue A-B: One-phase, Chained Transactions, Read-Only.

– Dialogue B-C: Commit, Chained Transactions, Read-Only.

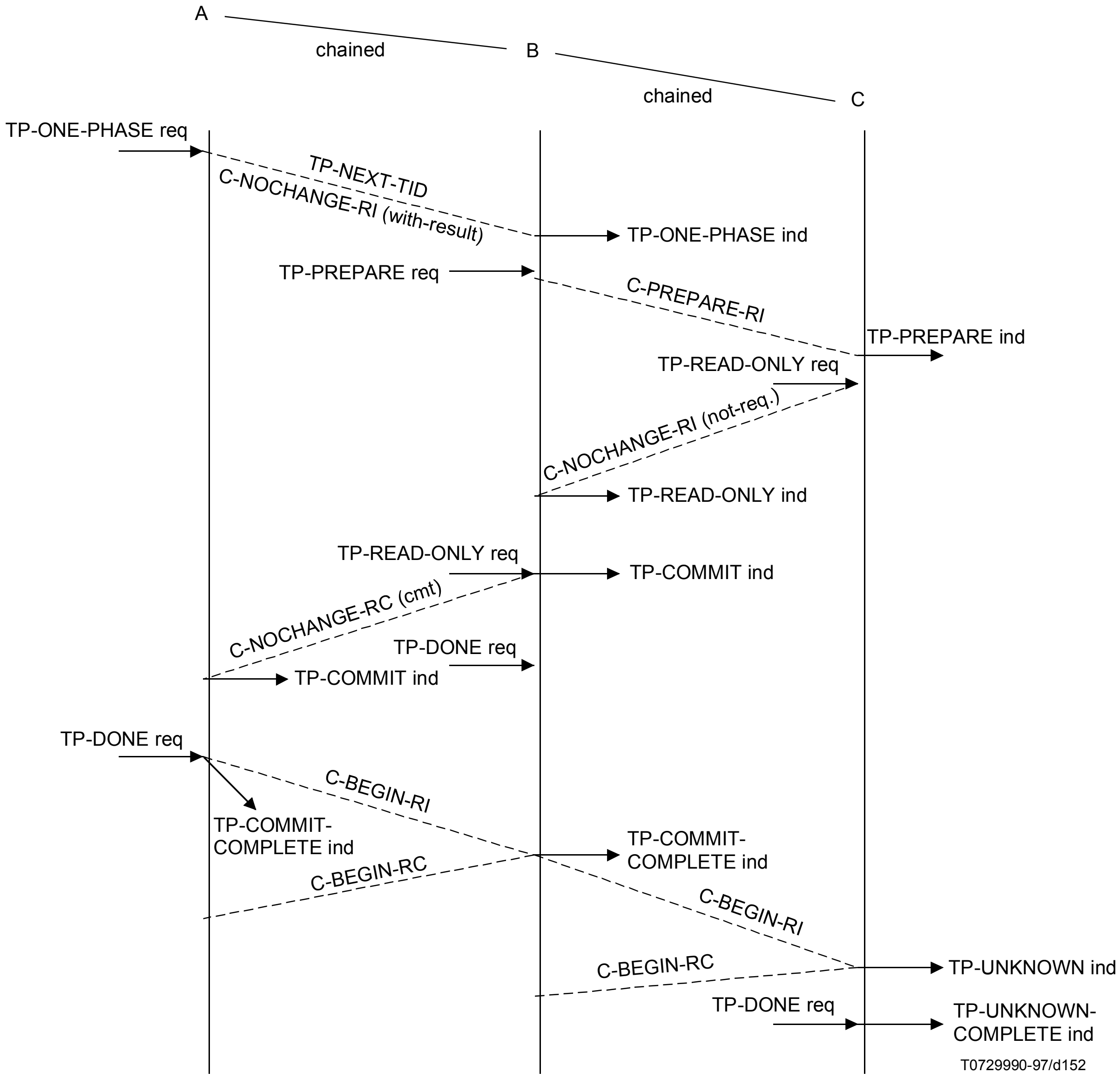


Figure C.146/X.862 – Static one-phase and read-only – Chained

C.13.12 Flow of C-BEGIN on static one-phase OCC-path – C-BEGIN not receivable

Node B issues C-COMMIT+C-BEGIN request after making the commit decision. C-BEGIN indication is awaited on the dialogue with the superior (static one-phase chaining, no reporting), after A-P-ABORT indication, the next transaction must be rolled back.

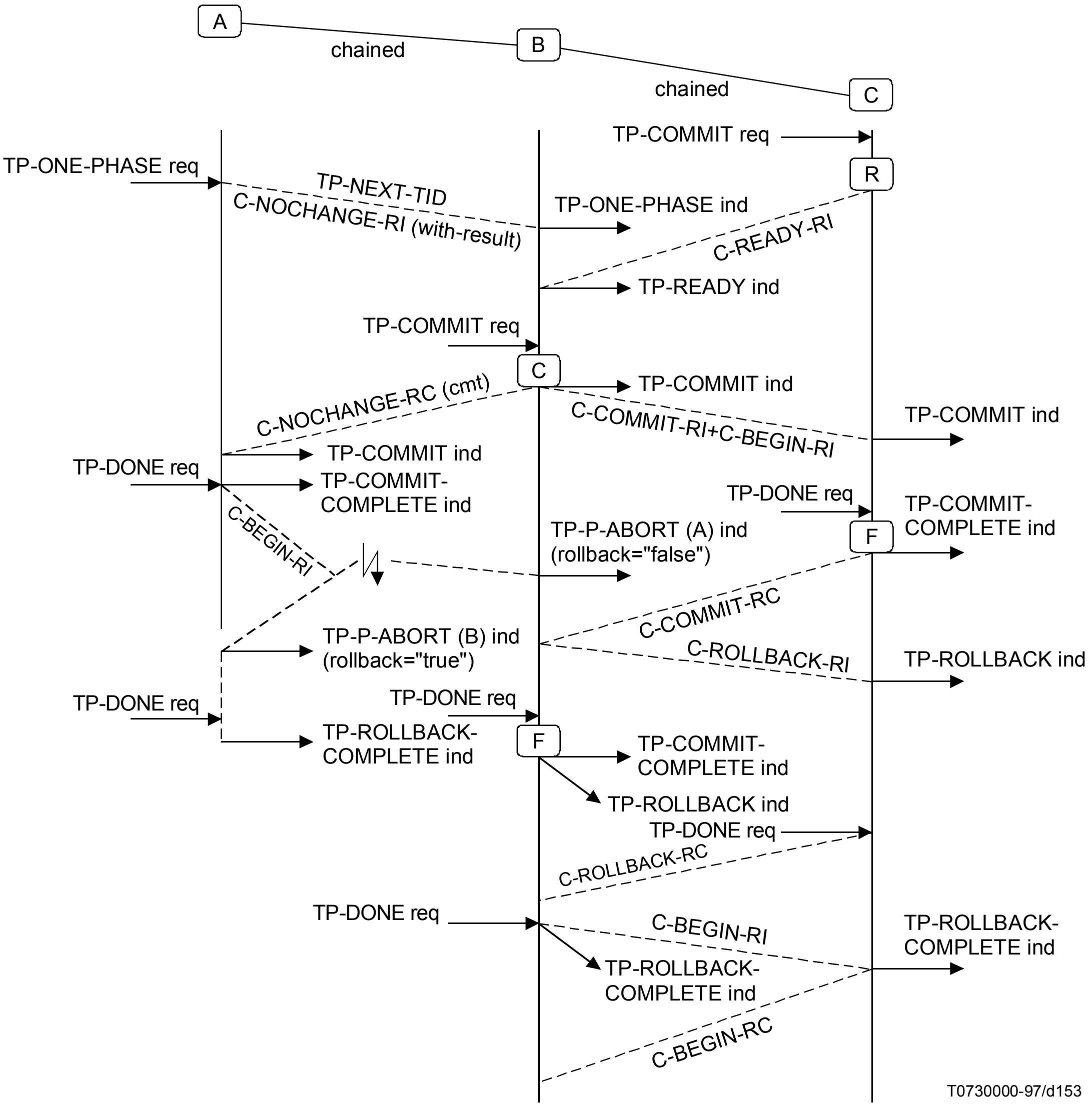


Figure C.147/X.862 – Flow of C-BEGIN on static one-phase OCC-path – C-BEGIN not receivable

C.13.13 Flow of C-BEGIN on static one-phase OCC-path – TP-U-ABORT pending

Node B issues C-COMMIT+C-BEGIN request after making the commit decision. C-BEGIN indication is awaited on the dialogue with the superior A (static one-phase chaining, no reporting), after TP-U-ABORT request, the next transaction must be rolled back (failure related actions are allowed after TP-P-ABORT of a non-chaining subordinate dialogue to node D, not shown in the scenario). After rollback is reported to the superior the dialogue is detached.

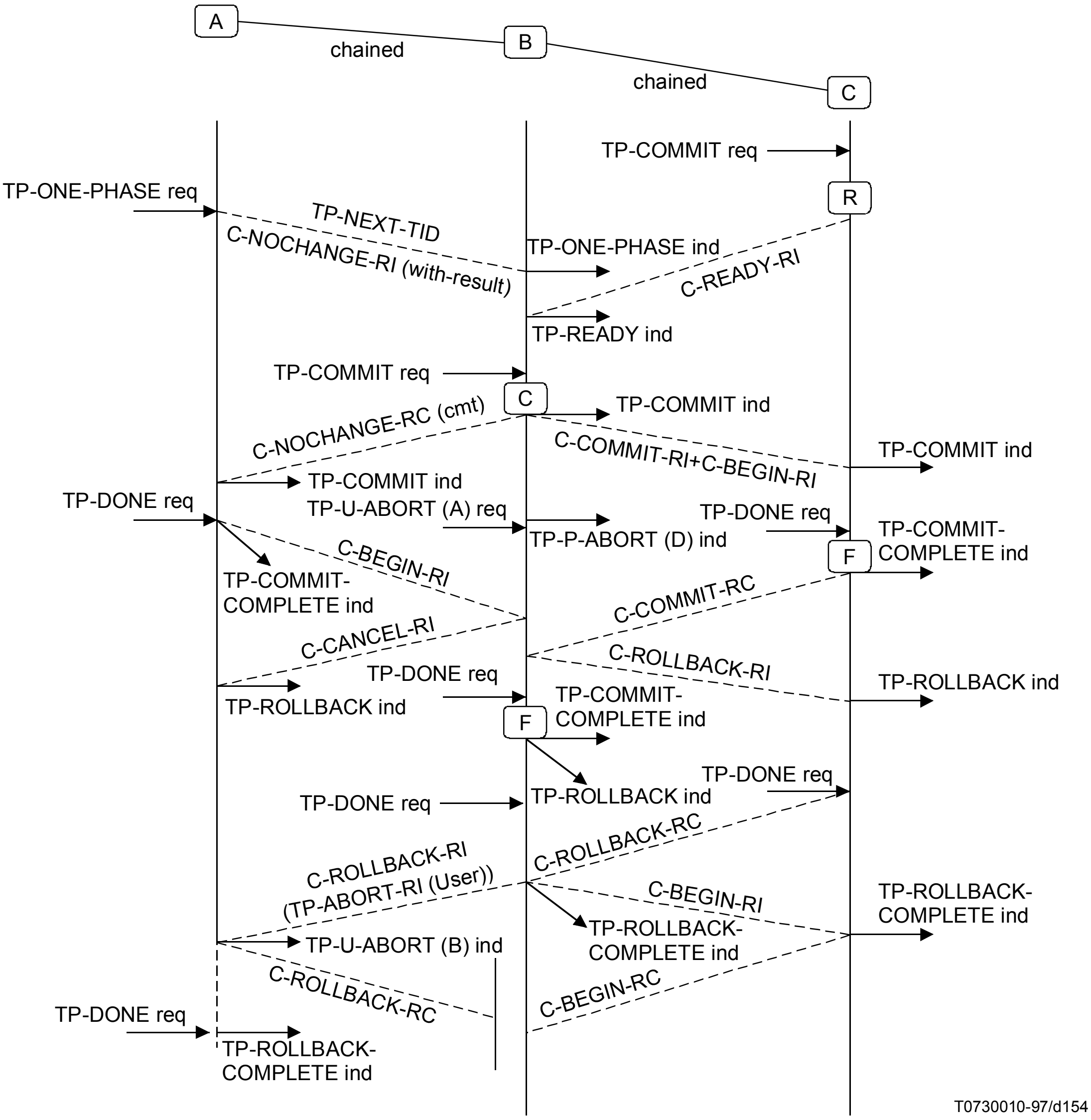


Figure C.148/X.862 – Flow of C-BEGIN on static one-phase OCC-path – TP-U-ABORT pending

C.13.14 Flow of C-BEGIN on static one-phase OCC-path – TP-U-ABORT request in Ready-state

Node B issues C-COMMIT+C-BEGIN request after receiving the decision from another subordinate (not shown in the scenario. C-BEGIN indication is not awaited on the dialogue with the superior (static one-phase chaining, no reporting), after TP-U-ABORT request. Node B is root of the transaction tree for the next transaction*, the identifier for the next transaction is created by node B*.

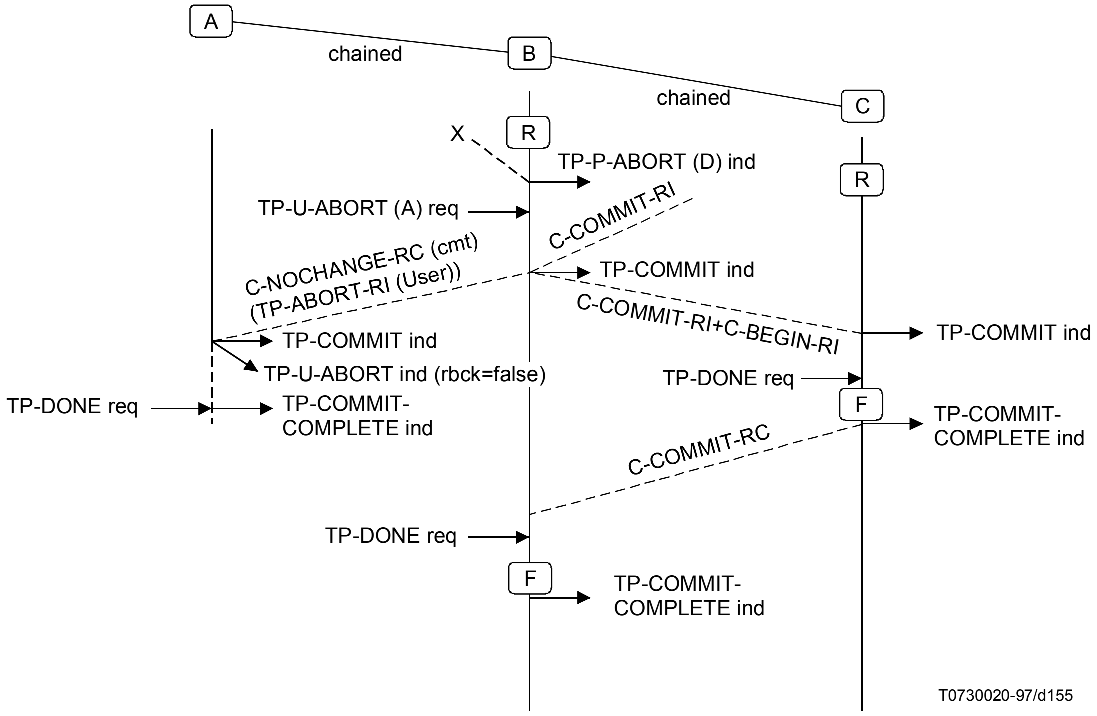


Figure C.149/X.862 – Flow of C-BEGIN on static one-phase OCC-path – TP-U-ABORT request in Ready-state

C.13.15 Flow of C-BEGIN on static one-phase OCC-path – TP-U-ABORT request in Ready-state

Node B issues C-COMMIT+C-BEGIN request after receiving the decision from another subordinate (no shown in the scenario. C-BEGIN indication is not awaited on the dialogue with the superior (static one-phase chaining, reporting), after TP-U-ABORT request. Node B is root of the transaction tree for the next transaction, *the identifier for the next transaction is created by node B*.

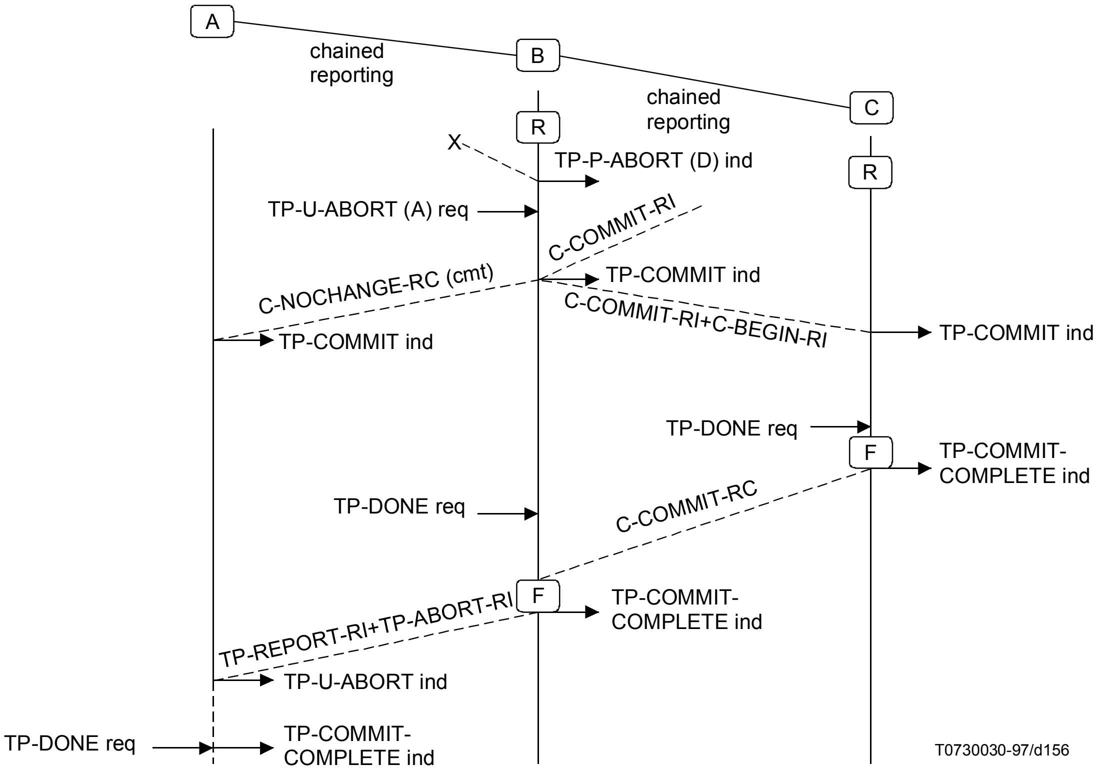


Figure C.150/X.862 – Flow of C-BEGIN on static one-phase OCC-path – TP-U-ABORT request in Ready-state

C.14 Implicit prepare scenarios

This subclause contains a number of scenarios which illustrate the use of the Implicit Prepare functional unit.

The subclause on dynamic commitment scenarios includes further scenarios with implicit prepare.

C.14.1 Implicit prepare with unchained transactions – Polarized control

The scenario of Figure C.151 shows the use of the Implicit prepare functional unit with the Commit, Unchained Transactions and Polarized Control functional units. The subordinate can tell from the application semantics that it can initiate commitment after processing of the request from the superior has been completed. A typical example would be a request/response application where a transaction can only contain one application request to which there can only be a single application response.

With Polarized Control, the superior must wait for the TP-READY indication before it can issue TP-COMMIT request as it does not have control of the dialogue.

The TP-COMMIT request at the subordinate could be replaced by a TP-READ-ONLY request or a TP-ONE-PHASE request if the appropriate functional units were also selected.

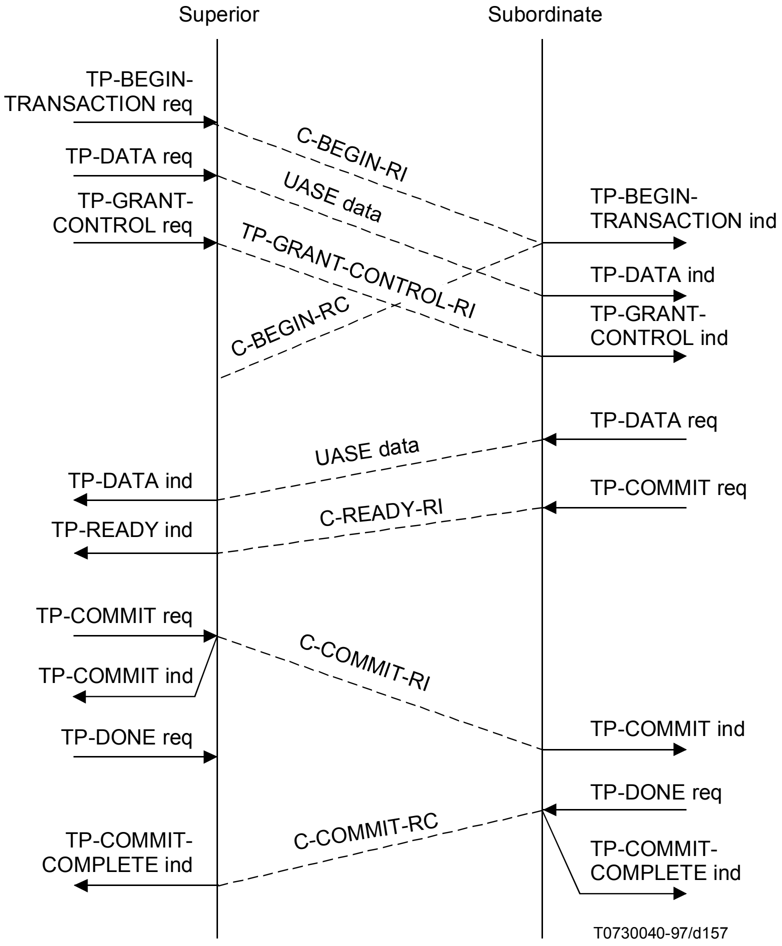


Figure C.151/X.862 – Implicit prepare with unchained transactions – Polarized control

C.14.2 Implicit prepare with unchained transactions – Shared control

The scenario of Figure C.152 shows the use of the Implicit prepare functional unit with the Commit, Unchained Transactions and Shared Control functional units. The subordinate can tell from the application semantics that it can initiate commitment after processing of the request from the superior has been completed. The superior initiates commitment after the reply has been received and does not need to wait for the TP-READY indication.

The TP-COMMIT request at the subordinate could be replaced by a TP-READ-ONLY request or a TP-ONE-PHASE request if the appropriate functional units were also selected.

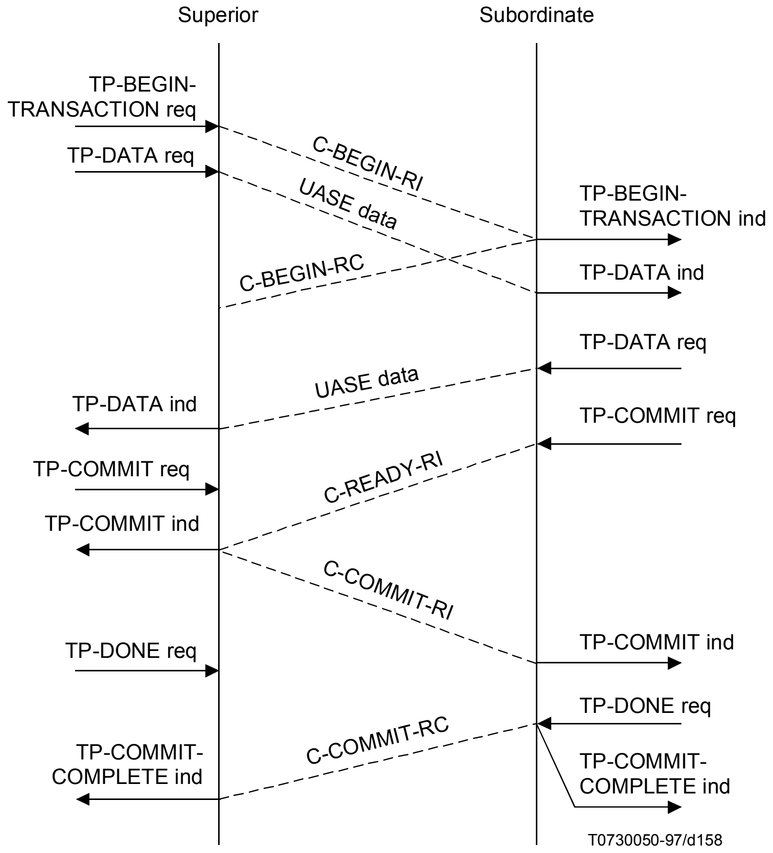


Figure C.152/X.862 – Implicit prepare with unchained transactions – Shared control

C.14.3 Implicit prepare with intermediate and read-only leaf – Chained transactions

The scenario of Figure C.153 shows the use of the Implicit prepare functional unit with the Commit, Chained Transactions, Read-Only and Shared Control functional units. The intermediate can tell from the application semantics that it can initiate commitment after processing of the request from the superior has been completed; the leaf reports read-only.

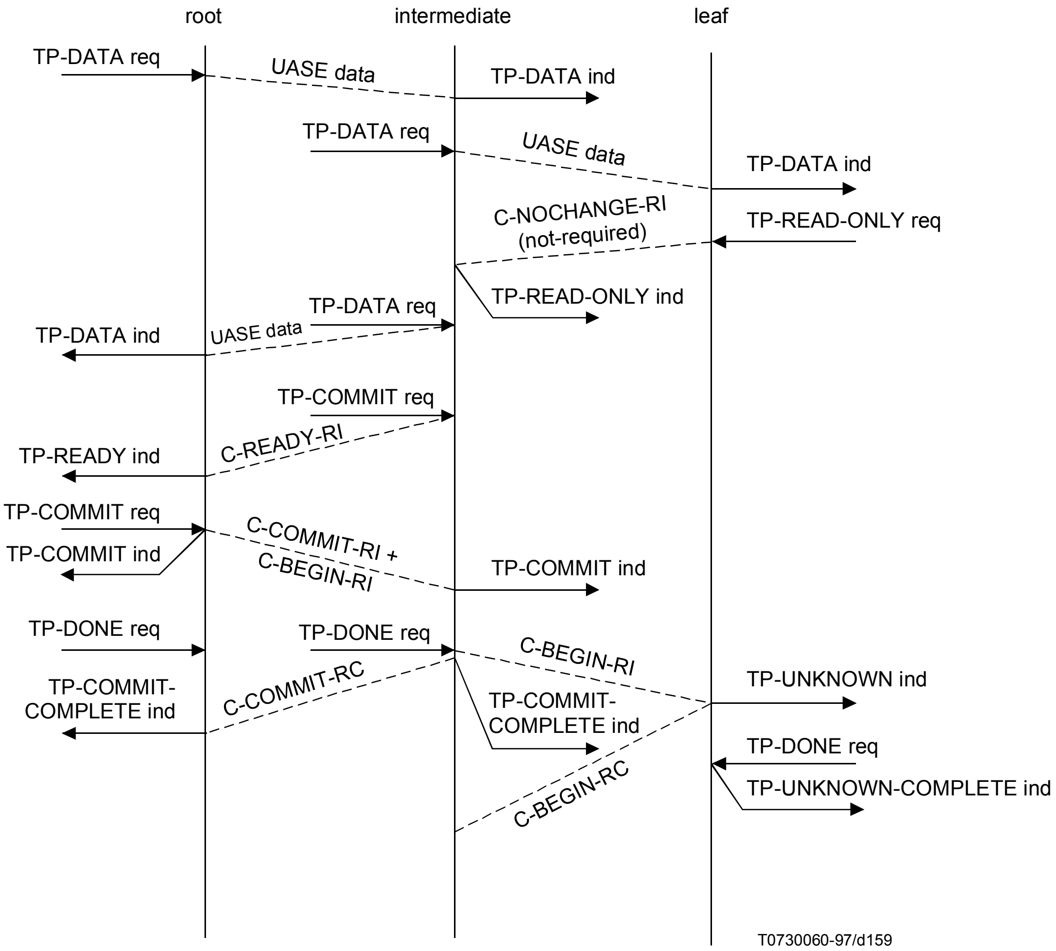


Figure C.153/X.862 – Implicit prepare with intermediate and read-only leaf – Chained transactions

C.14.4 Implicit prepare and heuristic commit

The scenario of Figure C.154 shows the use of the Implicit prepare functional unit with the Commit, Unchained Transactions and Shared Control functional units. The subordinate can tell from the application semantics that it can initiate commitment after processing of the request from the superior has been completed; the subordinate decides to make a heuristic commit decision. The superior initiates rollback and the damage report is received.

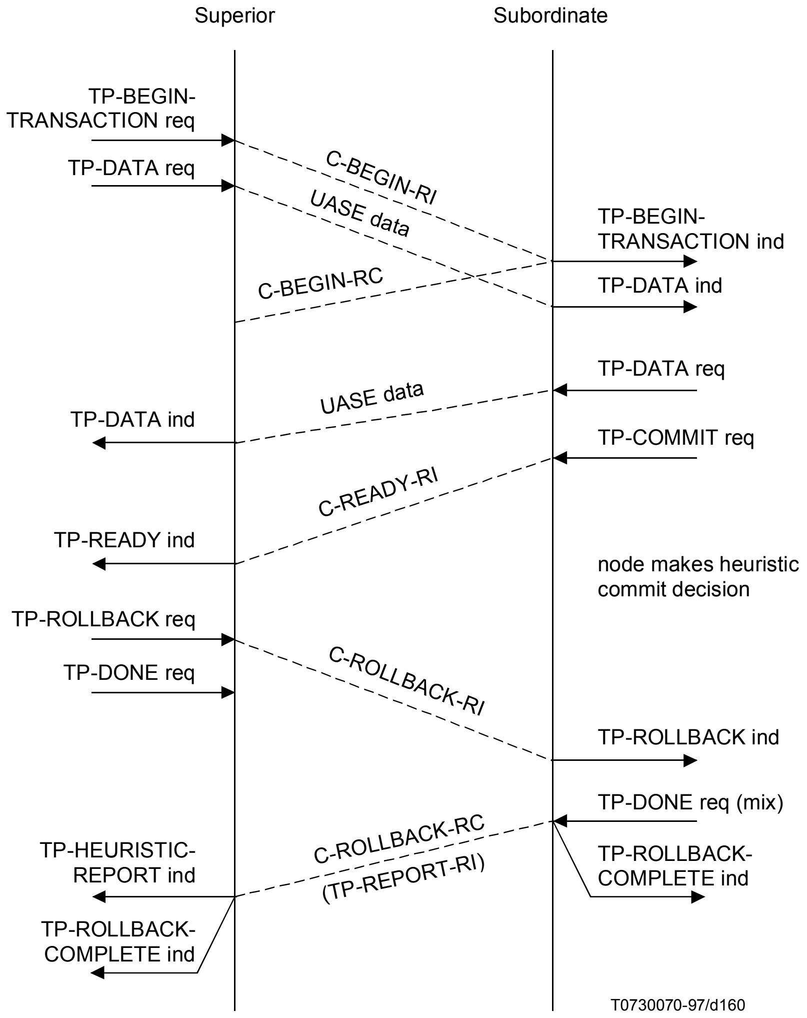


Figure C.154/X.862 – Implicit prepare and heuristic commit

C.14.5 Implicit prepare and heuristic rollback

The scenario of Figure C.155 shows the use of the Implicit prepare functional unit with the Commit, Unchained Transactions and Shared Control functional units. The subordinate can tell from the application semantics that it can initiate commitment after processing of the request from the superior has been completed; the subordinate decides to make a heuristic rollback decision. The superior initiates commitment and the damage report is received.

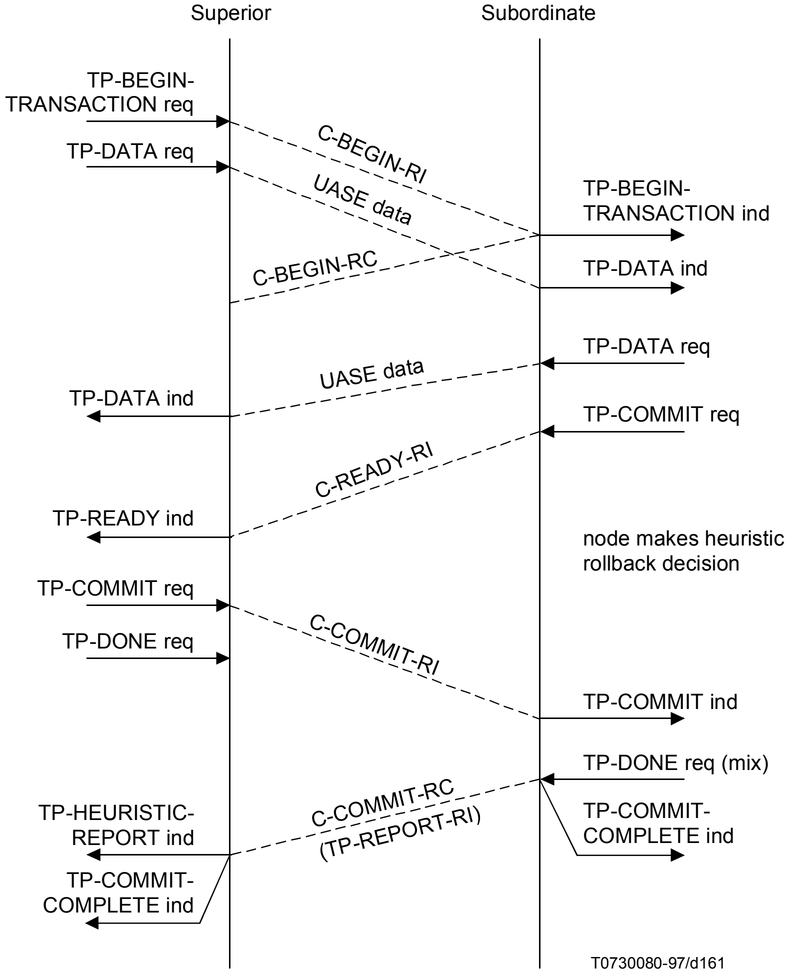


Figure C.155/X.862 – Implicit prepare and heuristic rollback

C.14.6 Implicit prepare, heuristic commit and dialogue abort

The scenario of Figure C.156 shows the use of the Implicit prepare functional unit with the Commit, Unchained Transactions and Shared Control functional units. The subordinate can tell from the application semantics that it can initiate commitment after processing of the request from the superior has been completed; the subordinate decides to make a heuristic commit decision. The dialogue to the superior aborts and the superior ceases to exist after rollback e. Recovery procedures do not find the superior and presume rollback; damage is reported by the subordinate TPSUI but there is no superior for it to be transmitted to.

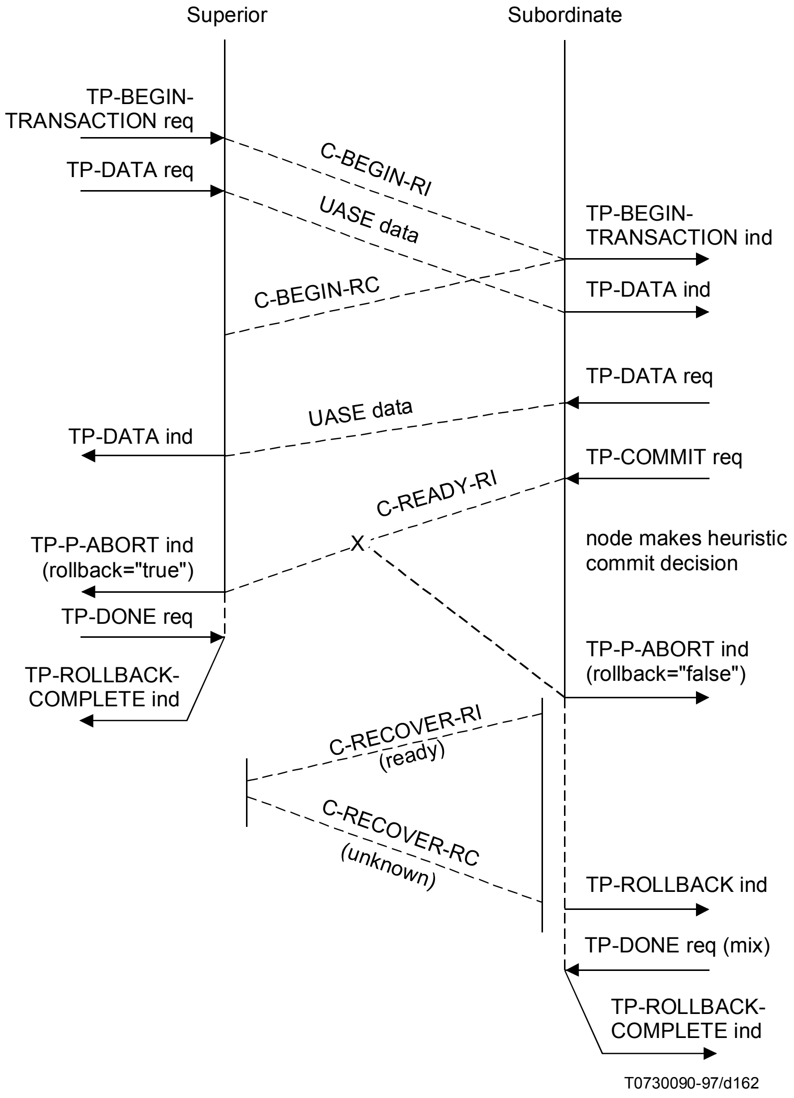


Figure C.156/X.862 – Implicit prepare, heuristic commit and dialogue abort

C.15 TP-ROLLBACK scenarios

A number of scenarios are presented showing rollback with a number of related functional units including Cancel, Heuristic Report Suppression, and Completion Diagnostics.

C.15.1 TP-ROLLBACK with Chained Transactions – Simple cases

C.15.1.1 Rollback from superior in active phase

The scenario of Figure C.157 (Figure C.16 from ITU-T Rec. X.862 (1993) | ISO/IEC 10026-3:1992) describes a sequence of primitives in the case when a transaction is rolled back by the superior TPSUI issuing a TP-ROLLBACK request during the active phase. The TPSUIs each immediately begin a new transaction.

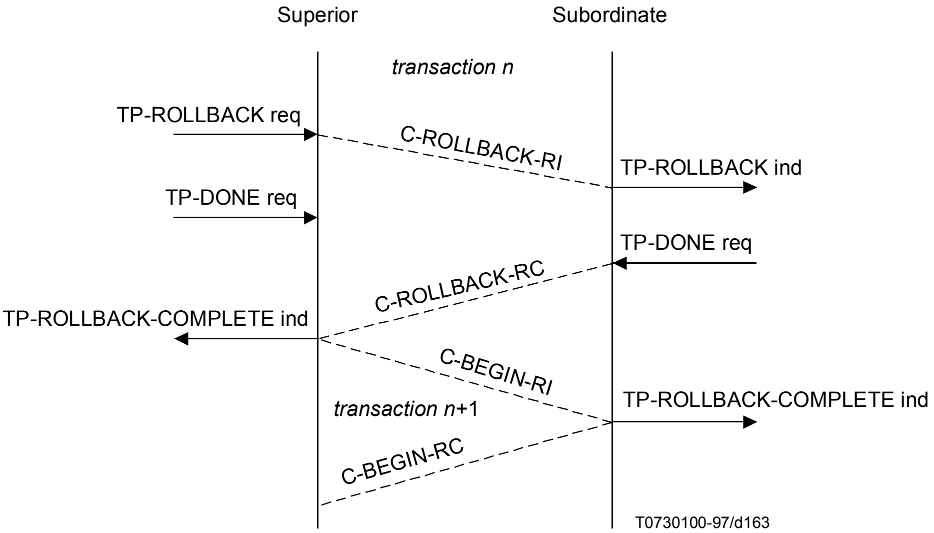


Figure C.157/X.862 – Rollback from superior in active phase

C.15.1.2 Rollback from subordinate in active phase

The scenario of Figure C.158 describes a sequence of primitives in the case when a transaction is rolled back by the subordinate TPSUI issuing a TP-ROLLBACK request during the active phase. The TPSUIs each immediately begin a new transaction.

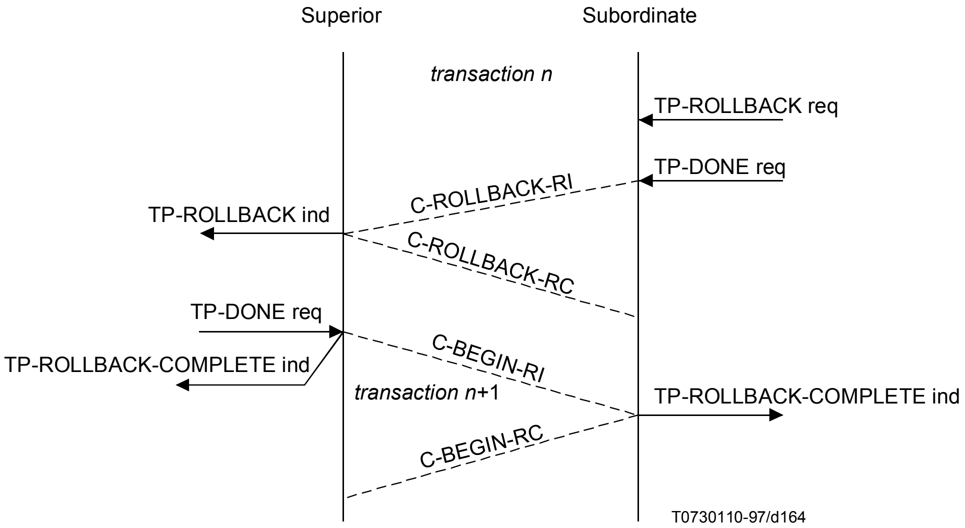


Figure C.158/X.862 – Rollback from subordinate in active phase

C.15.1.3 Rollback from both sides in active phase

The scenario of Figure C.159 describes a sequence of primitives in the case when a transaction is rolled back by both the superior and the subordinate TPSUI simultaneously issuing a TP-ROLLBACK request during the active phase; the collision is resolved by OSI Session (the C-ROLLBACK-RI issued by the responder of the underlying association will be purged). The TPSUIs each immediately begin a new transaction.

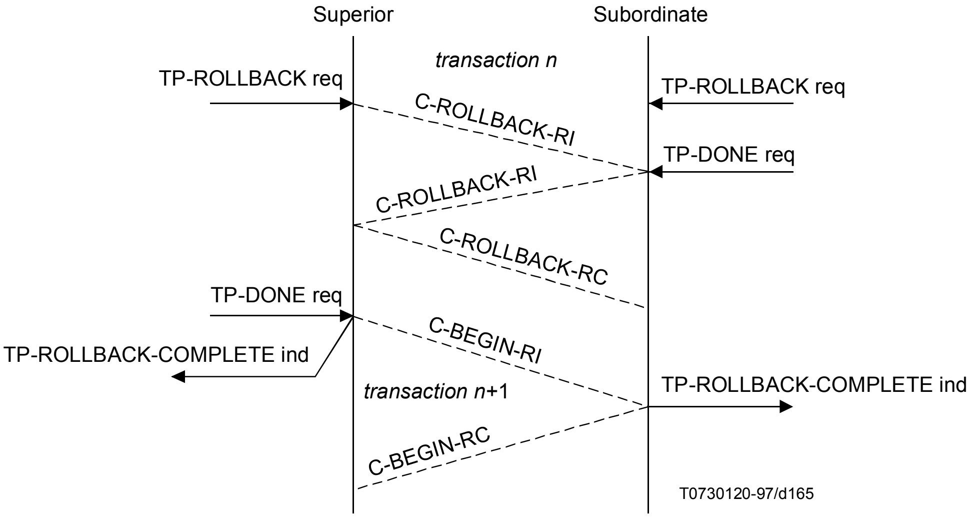


Figure C.159/X.862 – Rollback from both sides in active phase

C.15.1.4 Rollback in phase 1 of commitment

The scenario of Figure C.160 describes a sequence of primitives in the case when the subordinate reports ready, but the transaction is rolled back by the superior TPSUI issuing a TP-ROLLBACK request during the active phase. The TPSUIs each immediately begin a new transaction.

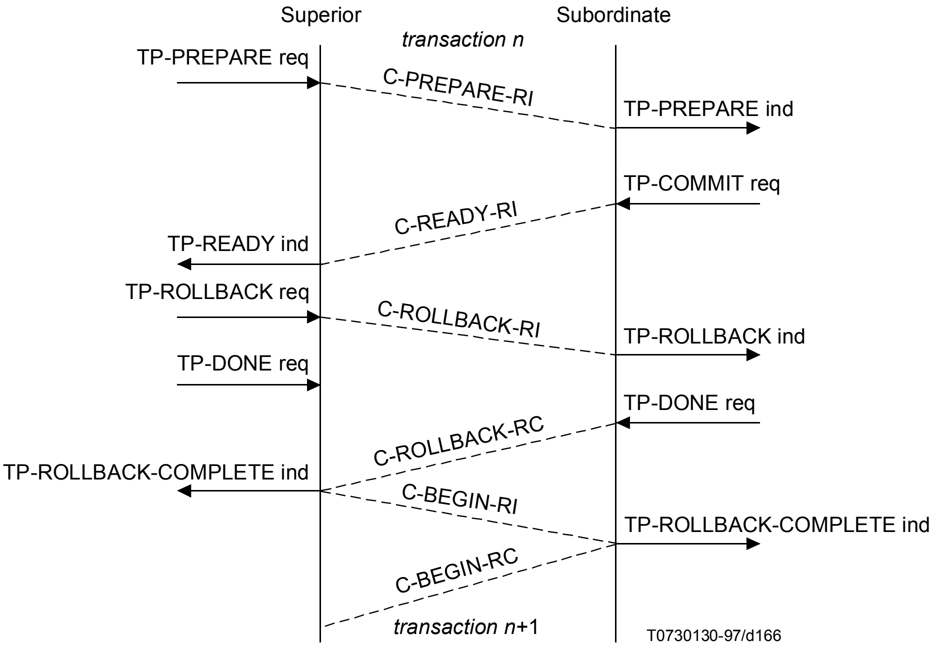


Figure C.160/X.862 – Rollback in phase 1 of commitment

C.15.1.5 Rollback in phase 1 of commitment – Heuristic damage is reported

The scenario of Figure C.161 describes a sequence of primitives in the case when the subordinate reports ready, but the transaction is rolled back by the superior TPSUI issuing a TP-ROLLBACK request during the active phase. However the subordinate made a heuristic decision which proves to be wrong; the damage is reported. The TPSUIs each immediately begin a new transaction.

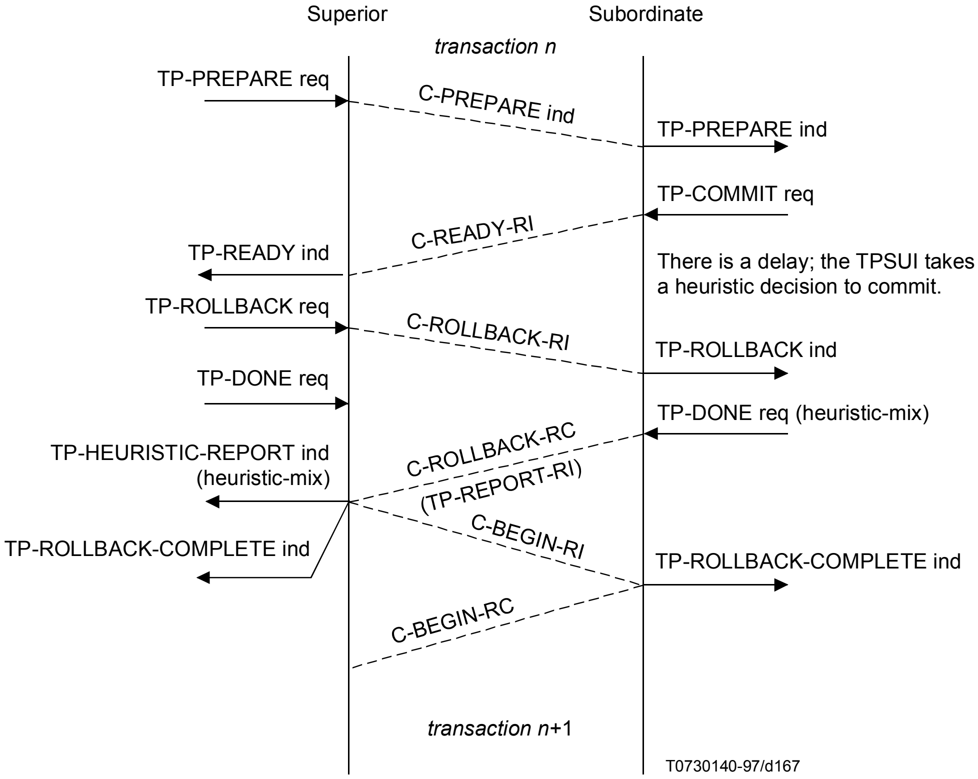


Figure C.161/X.862 – Rollback in phase 1 of commitment – Heuristic damage is reported

C.15.1.6 Rollback in phase 1 of commitment – Heuristic damage is not reported

The scenario of Figure C.162 describes a sequence of primitives in the case when the subordinate reports ready, but the transaction is rolled back by the superior TPSUI issuing a TP-ROLLBACK request during the active phase. The subordinate makes a heuristic decision which proves to be wrong; the Heuristic Report Suppression functional unit is selected and the damage is not reported to the superior. The TPSUIs each immediately begin a new transaction.

C.15.1.7 Rollback in phase 1 of commitment – A completion report is provided

The scenario of Figure C.163 describes a sequence of primitives in the case when the subordinate rolls back the transaction and provides a completion report. The Completion Diagnostics functional unit is selected. The TPSUIs each immediately begin a new transaction.

C.15.1.8 Rollback in phase 1 of commitment – A cancel is sent

The scenario of Figure C.164 describes a sequence of primitives in the case when the subordinate rolls back the transaction and provides a completion report. The Completion Diagnostics functional unit is selected. The Cancel functional unit is also selected and a cancel is sent to allow rollback to be accelerated. The TPSUIs each immediately begin a new transaction.

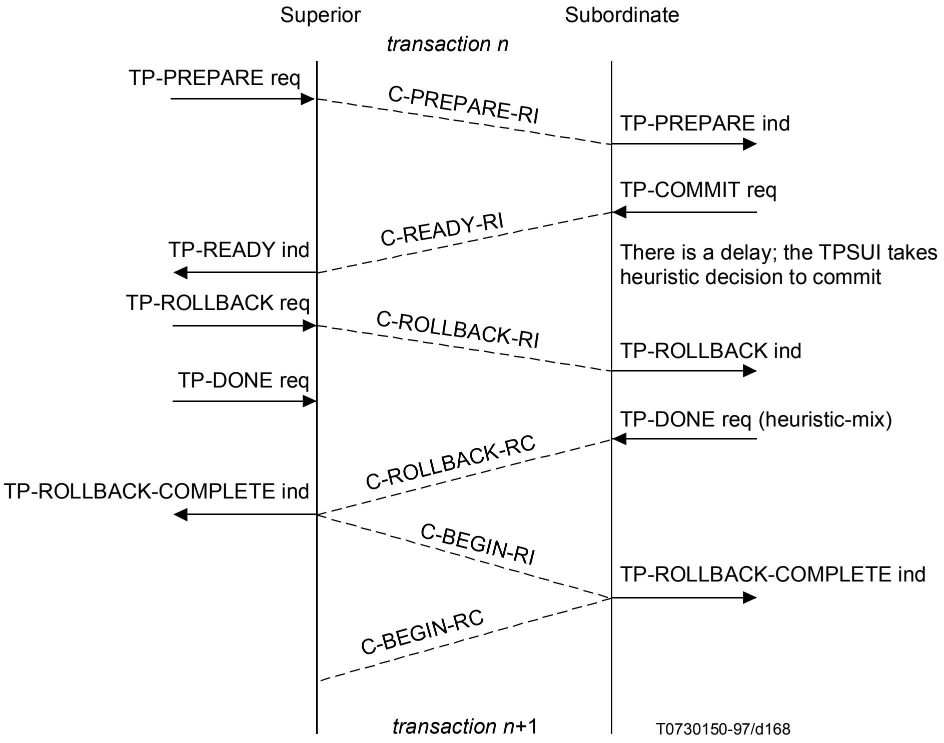


Figure C.162/X.862 – Rollback in phase 1 of commitment – Heuristic damage is not reported

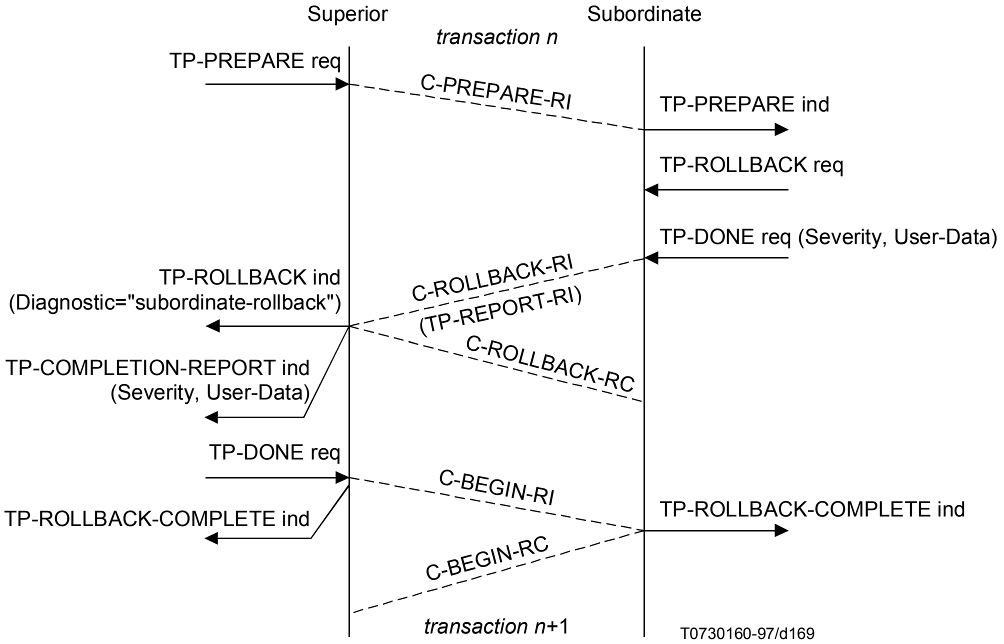


Figure C.163/X.862 – Rollback in phase 1 of commitment – A completion report is provided

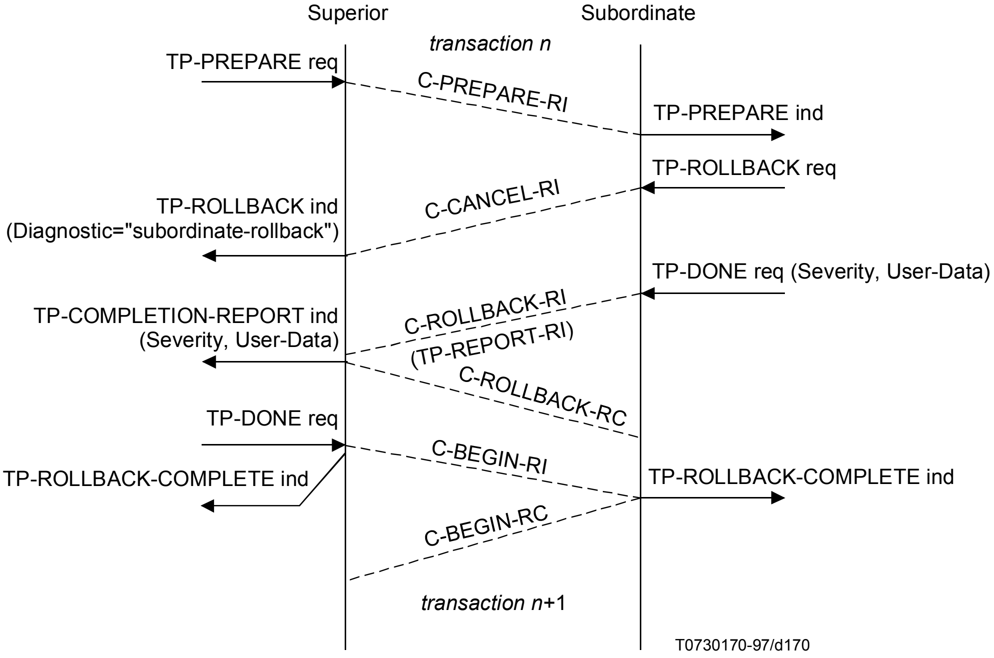


Figure C.164/X.862 – Rollback in phase 1 of commitment – A cancel is sent

C.15.2 TP-ROLLBACK with Unchained Transactions – Simple cases

C.15.2.1 Rollback from superior in active phase

The scenario for TP-ROLLBACK with unchained transactions, shown in Figure C.17 from ITU-T Rec. X.862 (1993) | ISO/IEC 10026-3:1992, is identical to the scenario for chained transactions except that a new transaction will not begin until a TP-BEGIN-TRANSACTION request has been issued.

C.15.2.2 Rollback from subordinate in active phase

The scenario of Figure C.166 describes a sequence of primitives in the case when a transaction is rolled back by the subordinate TPSUI issuing a TP-ROLLBACK request during the active phase.

C.15.3 TP-ROLLBACK with Unchained Transactions – Complex cases

C.15.3.1 Rollback in phase 1 of commitment – Completion reports are provided

The scenario of Figure C.167 describes a sequence of primitives in the case when two leaf nodes each initiate rollback of the transaction and provide a completion report; the intermediate TPSUI reacts to each completion report and finally a report is issued to the root node. The Completion Diagnostics functional unit is selected and the Unchained Transactions case is shown.

In the scenario, the arrival of a C-ROLLBACK-RI from D would immediately be propagated to C as shown by "C-RBCK-RI**"; to avoid cluttering the diagram further, the reader is asked to imagine that this collides with the C-ROLLBACK-RI coming from C and is discarded.

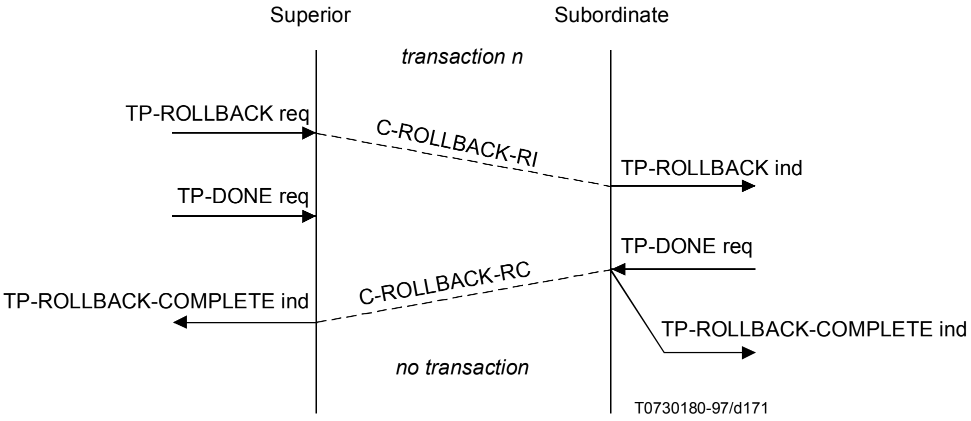


Figure C.165/X.862 – Rollback from superior in active phase

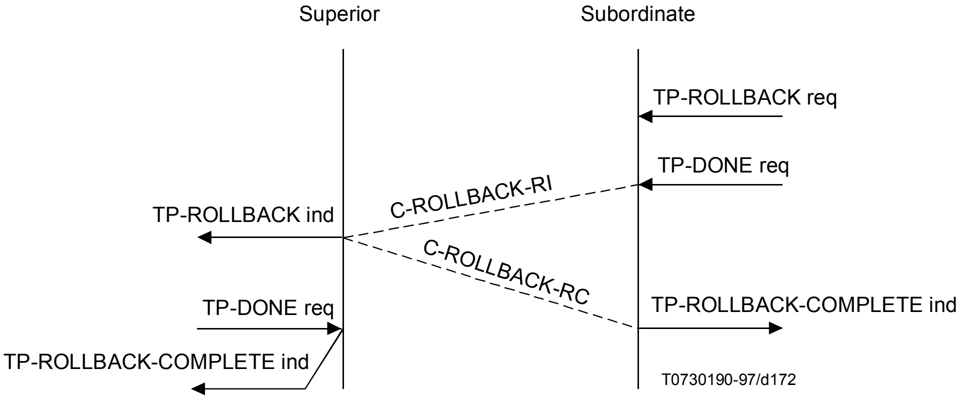


Figure C.166/X.862 – Rollback from subordinate in active phase

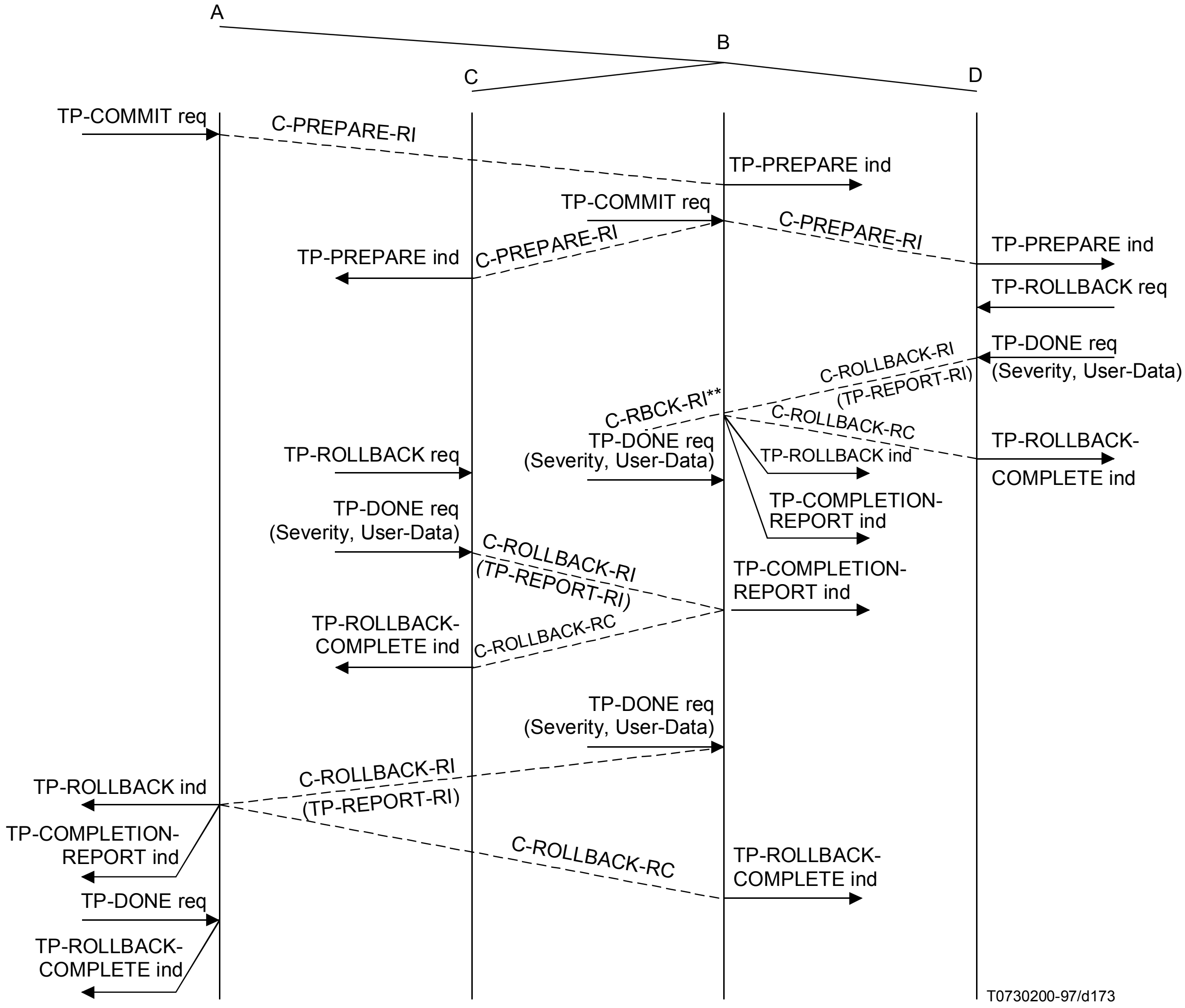


Figure C.167/X.862 – Rollback in phase 1 of commitment – A completion report is provided

C.15.3.2 Rollback in phase 1 of commitment, cancel is used

As can be seen from the previous scenario, there can be a significant delay in reporting up the tree that the transaction is being rolled back; other parts of the tree will continue to consume resources processing a transaction which will inevitably be rolled back. This next scenario describes a sequence of primitives in the case when two leaf nodes each initiate rollback of the transaction and provide a completion report; the intermediate TPSUI reacts to each completion report and finally a report is issued to the root node. However the cancel is sent as early as possible to warn the root of the pending rollback. The Cancel and Completion Diagnostics functional units are selected and the Unchained Transactions case is shown.

In the scenario, the arrival of a C-CANCEL-RI at B from D would immediately be propagated to C as shown by "C-RBCK-RI**"; to avoid cluttering the diagram further, the reader is asked to imagine that this firstly causes the "C-CANCEL-RI**" from C to be purged, but then collides with the C-ROLLBACK-RI coming from C and is in turn purged.

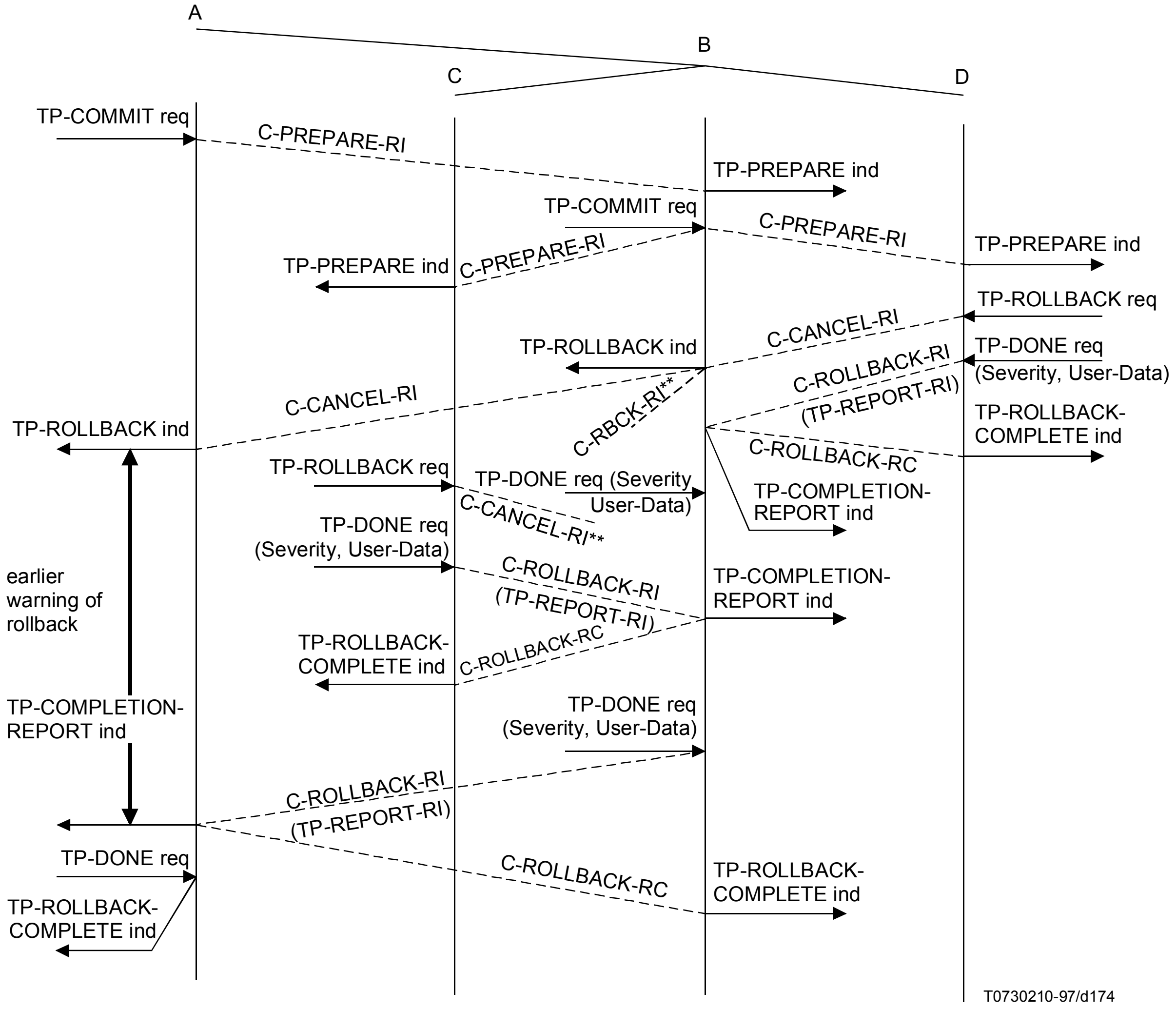


Figure C.168/X.862 – Rollback in phase 1 of commitment – Cancel is used

C.16 Dynamic Commitment scenarios

Dynamic commitment is only supported with unchained transactions; so all scenarios following are for unchained transactions even if this is not specified explicitly.

The symbol in figure C.169 which appears in the headings of the scenarios following indicates in which direction ready can flow on the dialogue (as set by parameters on the TP-BEGIN-DIALOGUE request and indication).

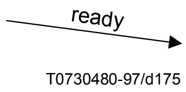


Figure C.169/X.862 – Ready direction symbol

C.16.1 Simple scenarios with two nodes

C.16.1.1 Ready down tree with TP-COMMIT request at subordinate

The dialogue controls are set so that READY or READY substitutes may not flow up tree.

Functional units selected: Commit, Dynamic Commit, Unchained Transactions.

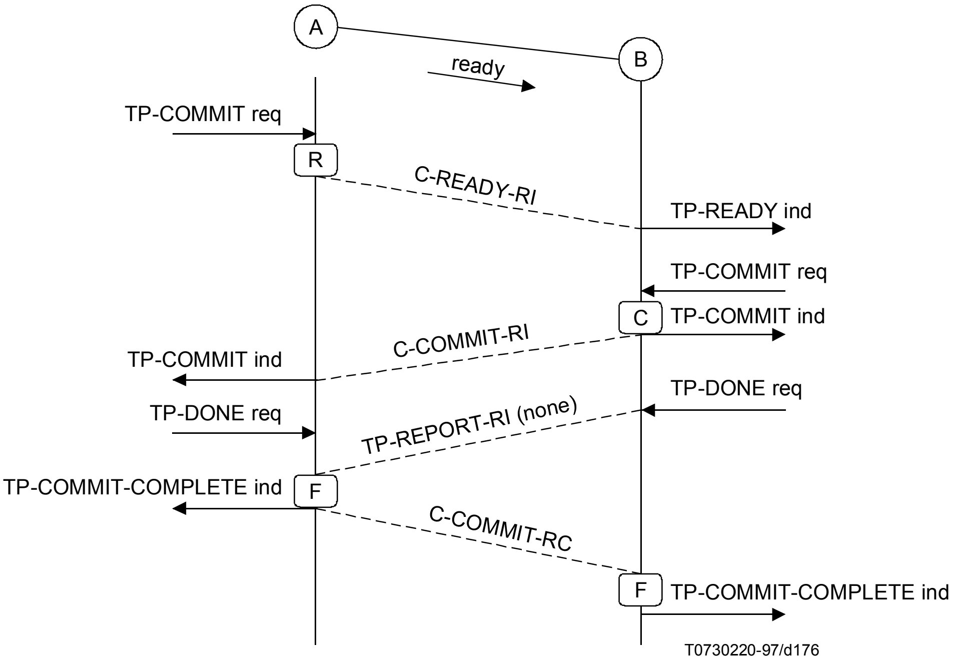


Figure C.170/X.862 – Ready down tree with TP-COMMIT request at subordinate

C.16.1.2 Ready down tree with TP-READ-ONLY request at subordinate which becomes coordinator

The dialogue controls are set so that READY or READY substitutes may not flow up tree. So although the subordinate TPSUI issues a TP-READ-ONLY request, the subordinate TPPM is forced to become the commit coordinator.

Functional units selected: Commit, Dynamic Commit, Read Only, Unchained Transactions.

A similar scenario and constraints would apply if the subordinate issued a TP-ONE-PHASE request.

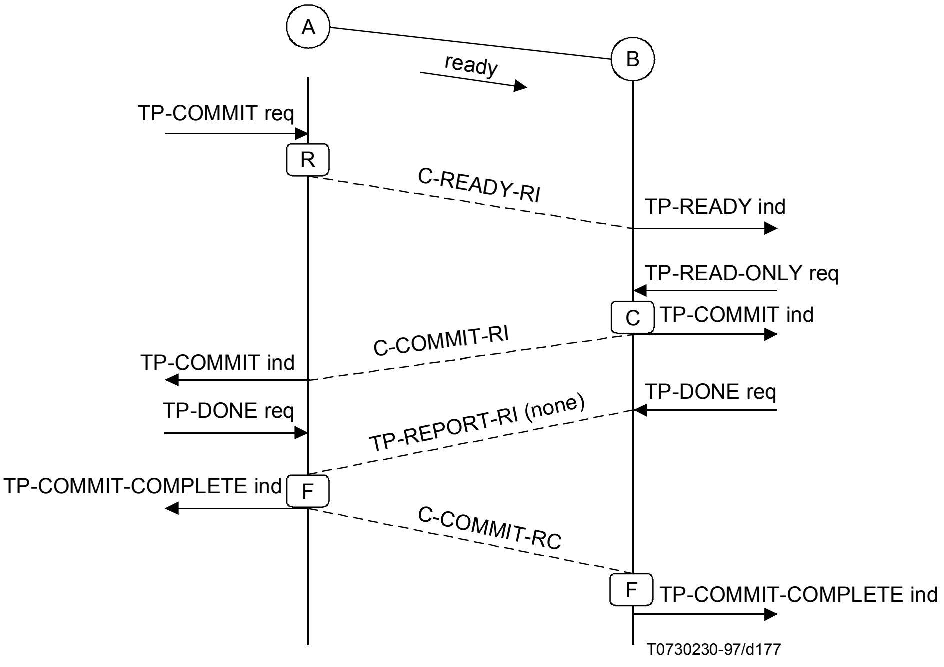


Figure C.171/X.862 – Ready down tree with TP-READ-ONLY request at subordinate which becomes coordinator

C.16.1.3 Ready down tree with TP-READ-ONLY request at subordinate which refuses to become coordinator

The dialogue controls are set so that READY or READY substitutes may up or down the tree. So when the subordinate TPSUI issues a TP-READ-ONLY request, the subordinate TPPM ignores the receipt of the C-READY-RI from the superior, sends a C-NOCHANGE-RI, and forces the superior to become the commit coordinator.

Functional units selected: Commit, Dynamic Commit, Read Only, Unchained Transactions.

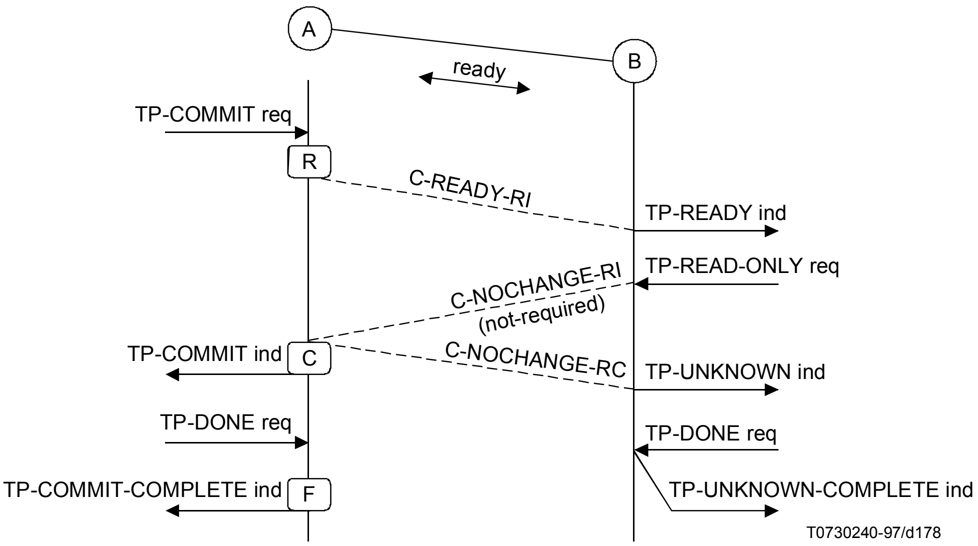


Figure C.172/X.862 – Ready down tree with TP-READ-ONLY request at subordinate which refuses to become coordinator

C.16.1.4 Ready down tree with TP-ONE-PHASE request at subordinate

The dialogue controls are set so that READY or READY substitutes may up or down the tree. So when the subordinate TPSUI issues a TP-ONE-PHASE request, the subordinate TPPM ignores the receipt of the C-READY-RI from the superior, sends a C-NOCHANGE-RI, and forces the superior to become the commit coordinator.

Functional units selected: Commit, Dynamic Commit, One phase, Unchained Transactions.

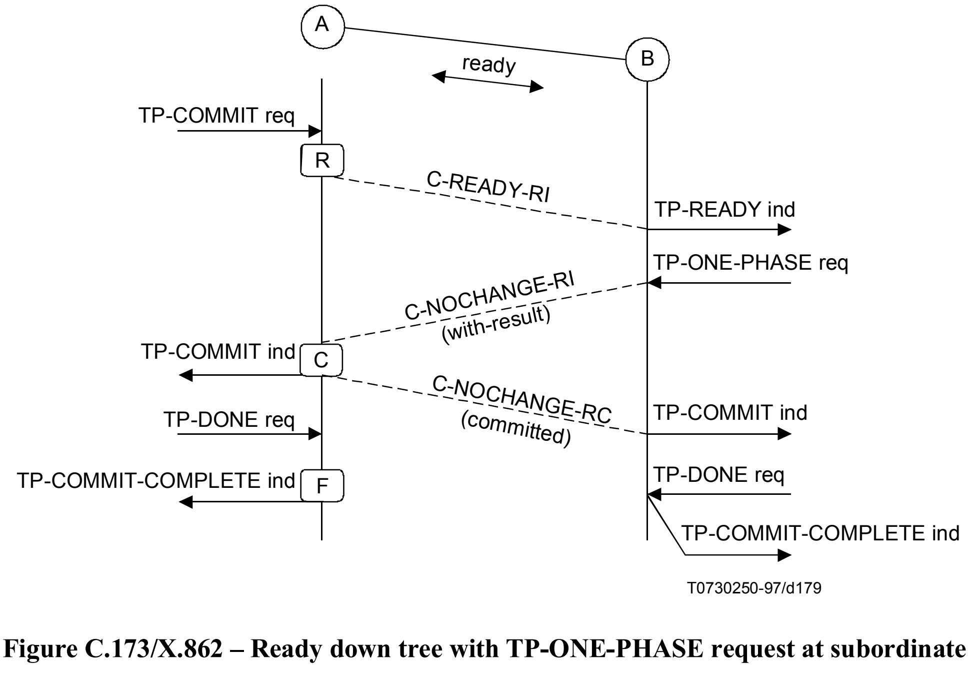


Figure C.173/X.862 – Ready down tree with TP-ONE-PHASE request at subordinate

C.16.2 Polarized control and dynamic commitment

The nature of polarized control places some constraints on the full use of some dynamic commit features. Generally in the scenarios following, whether Shared Control or Polarized Control functional units are selected is not specified unless this has a significant effect on the applicability of the scenario. The following scenarios illustrate some additional effects.

C.16.2.1 Ready down tree only – Subordinate has control

Functional units selected on A-B: Commit, Dynamic Commit, Implicit Prepare, Unchained Transactions.

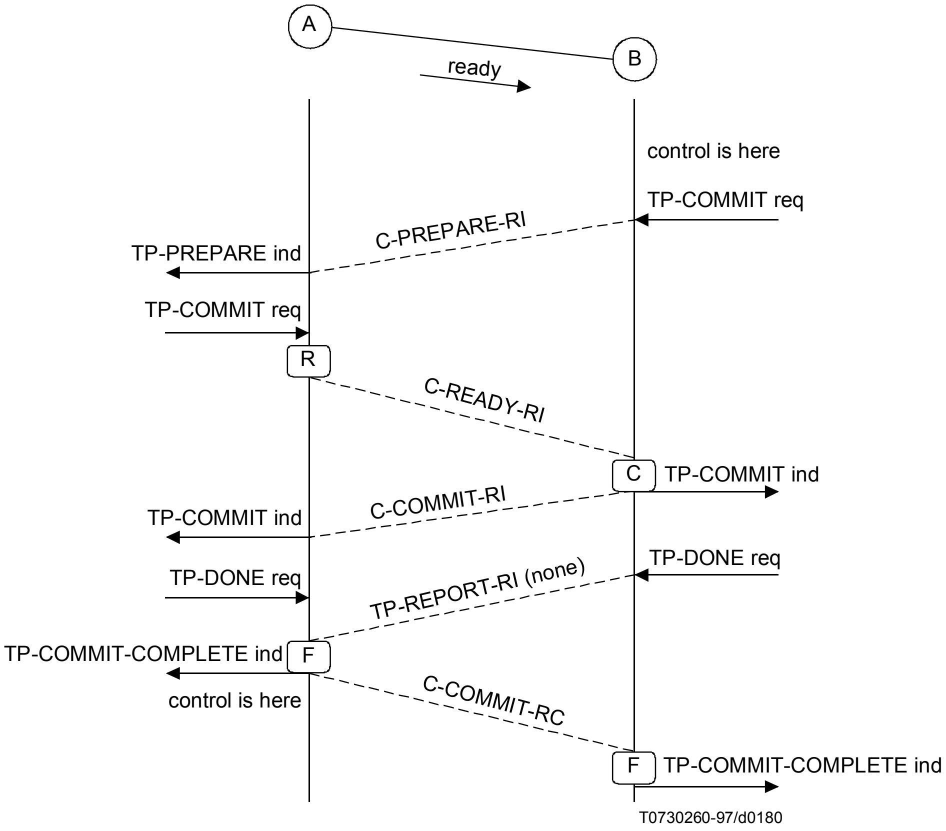


Figure C.174/X.862 – Ready down tree only – Subordinate has control

C.16.2.2 Ready either way – Subordinate becomes coordinator

The subordinate B initiates commitment; as it has multiple neighbours and ready can be received from each, prepares are sent to each. The subordinate of B replies read-only; B enters the ready state and sends ready. There is a ready/ready collision and the tie-break is used to select B as the commit coordinator (details in clause 8).

Functional units selected on A-B: Commit, Dynamic Commit, Implicit Prepare, Unchained Transactions.

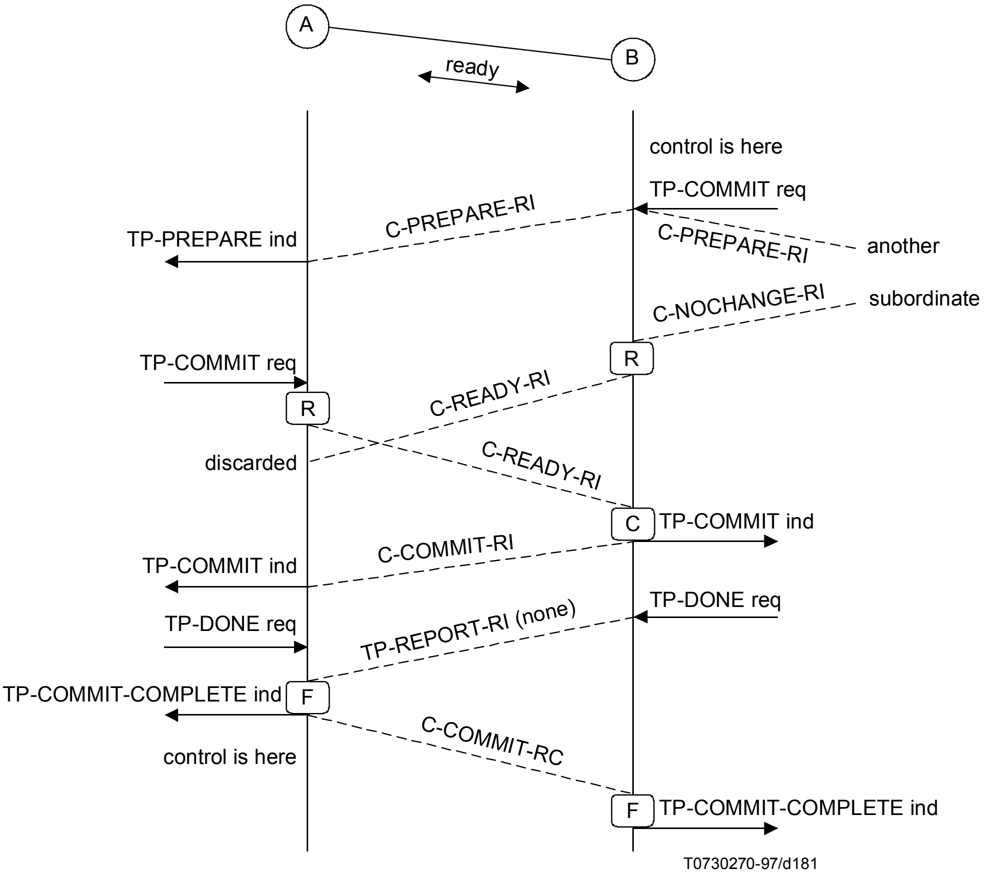


Figure C.175/X.862 – Ready either way – Subordinate becomes coordinator

C.16.2.3 TP-PREPARE with Data-Permitted set to "true", followed by TP-PREPARE

The superior has control and issues a TP-PREPARE request with Data-Permitted set to "true"; the subordinate returns data and the issues a TP-PREPARE request to signal that no more data will be sent and to allow the superior to initiate commitment.

Functional units selected on A-B: Commit, Dynamic Commit, Unchained Transactions.

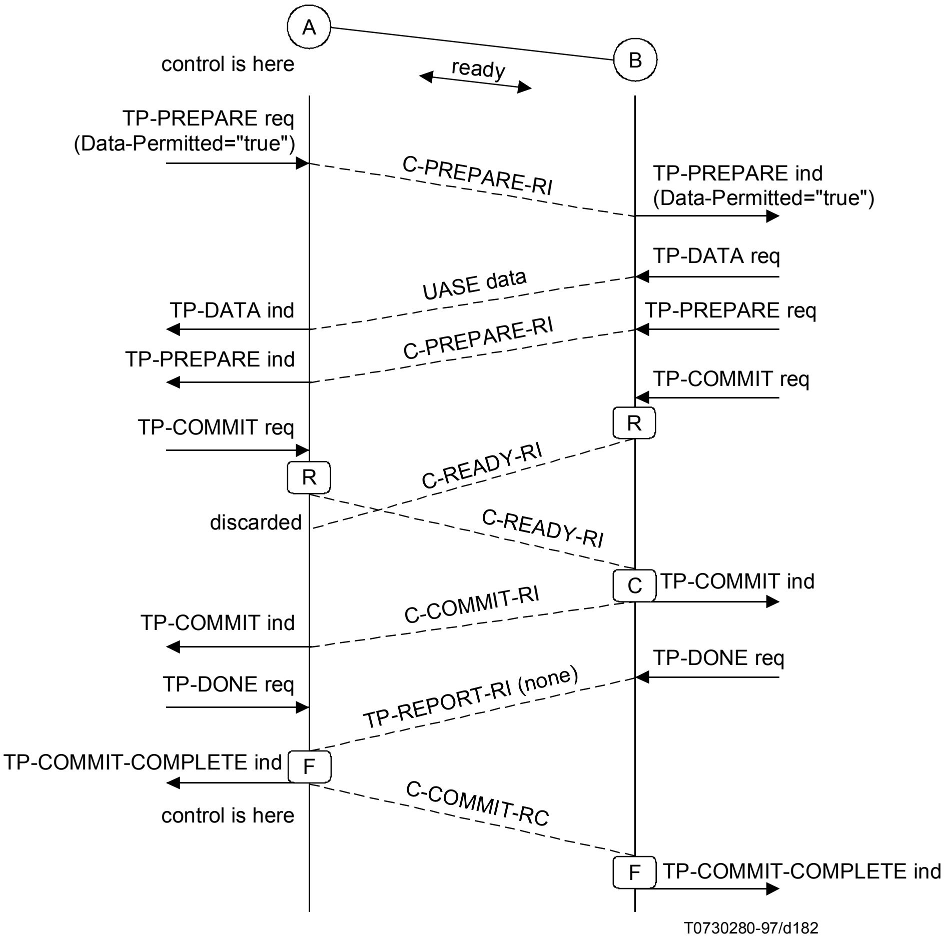


Figure C.176/X.862 – TP-PREPARE with Data-Permitted set to "true", followed by TP-PREPARE

C.16.2.4 Implicit prepare with dynamic one-phase commit

The scenario of Figure C.177 shows the use of implicit prepare with dynamic one-phase. The subordinate can tell from the application semantics that it can initiate commitment after processing of the request from the superior has been completed.

Functional units selected on A-B: Commit, Dynamic Commit, One-phase, Implicit Prepare, Unchained Transactions.

With Polarized Control, the superior must wait for the TP-PREPARE indication before it can issue TP-ONE-PHASE request as it does not have control of the dialogue.

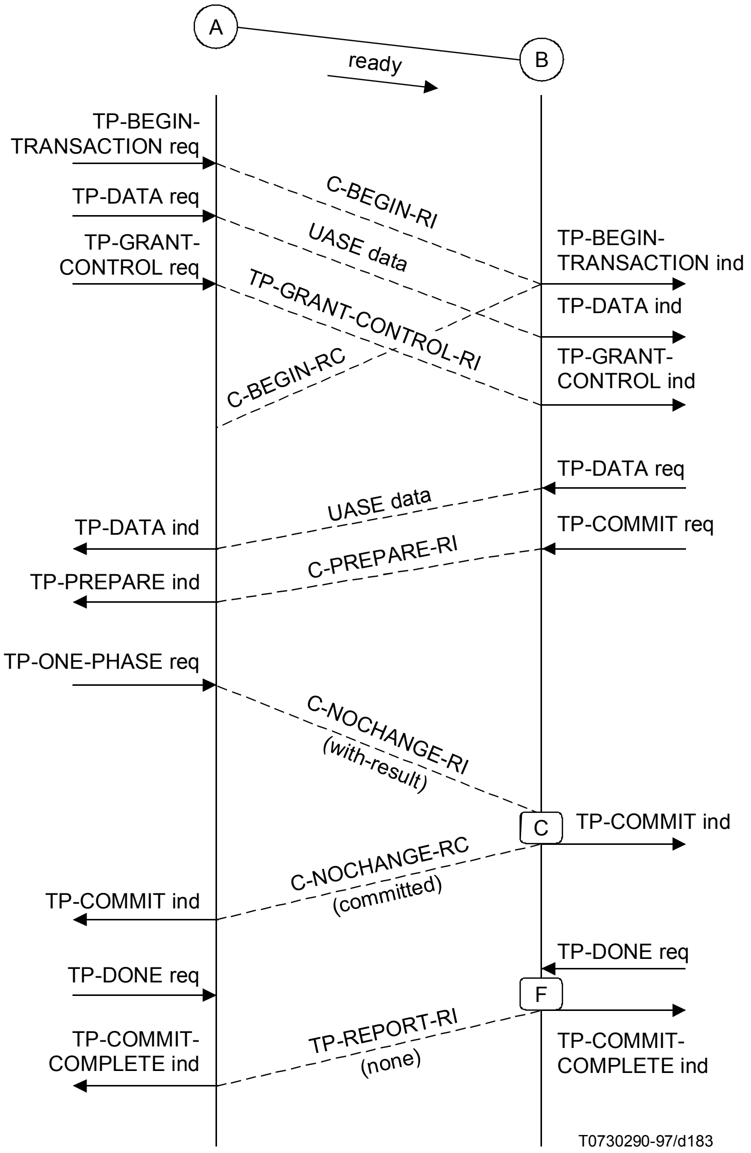


Figure C.177/X.862 – Implicit prepare with dynamic one-phase commit

C.16.3 Collisions with two nodes

Collision of C-READY-RI and C-READY-RI is described in clause 8 including the tie-break mechanism.

Collision of C-READY-RI with C-NOCHANGE-RI is also described in clause 8.

C.16.3.1 Collision of ONE-PHASE with ONE-PHASE

In effect the C-NOCHANGE-RI PDUs act as the confirmation for each other.

Functional units selected as a minimum: Commit, Dynamic Commit, One phase, Unchained Transactions.

In addition, the scenario is either Shared Control and the Implicit Prepare functional unit is selected, or a TP-PREPARE request/indication was previously issued.

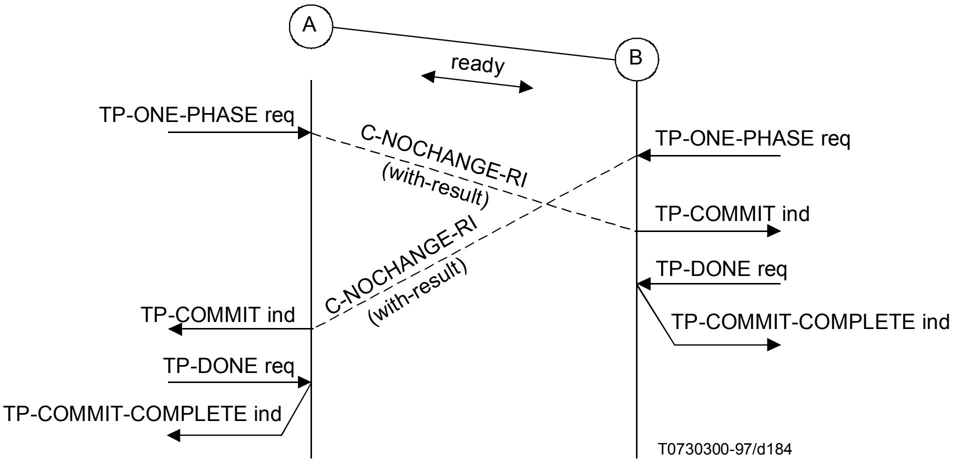


Figure C.178/X.862 – Collision of ONE-PHASE with ONE-PHASE

C.16.3.2 Collision of ONE-PHASE with READ-ONLY

The result of this collision is similar.

Functional units selected as a minimum: Commit, Dynamic Commit, One phase, Read Only, Unchained Transactions.

In addition, the scenario of Figure C.179 is either Shared Control and the Implicit Prepare functional unit is selected, or a TP-PREPARE request/indication was previously issued.

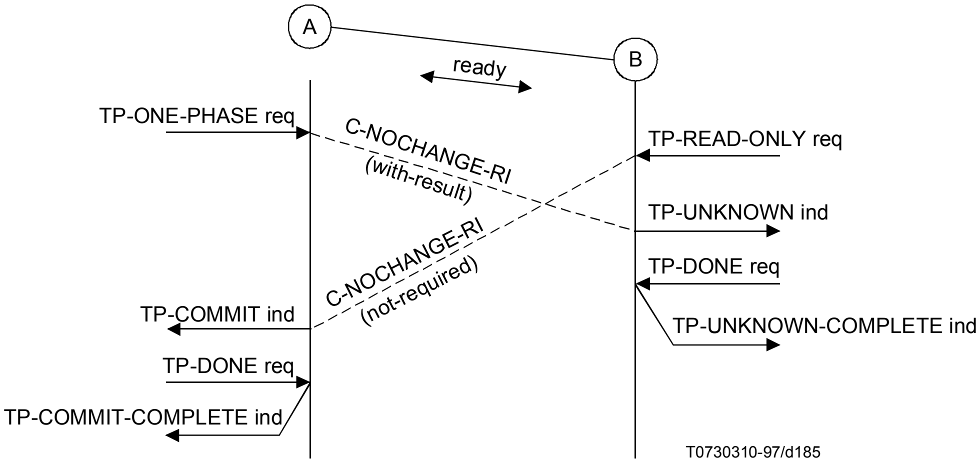


Figure C.179/X.862 – Collision of ONE-PHASE with READ-ONLY

C.16.4 Alternate commit initiator

The scenario of Figure C.180 shows commitment being initiated from a non-root node. This is sometimes known as an "alternate commit initiator" i.e. not the root node.

As a minimum the following functional units are selected:

– Dialogue A-C: Commit, Dynamic Commit, Unchained Transactions, Heuristic Report Suppression, Implicit Prepare.

– Dialogue A-B: Commit, Unchained Transactions. (B could be a implementation of OSI TP 1992.)

An alternative scenario would have the root A write the optional commit-log on receipt of the C-COMMIT-RI and immediately return a C-COMMIT-RC to C to inform it that its recovery responsibilities had ended.

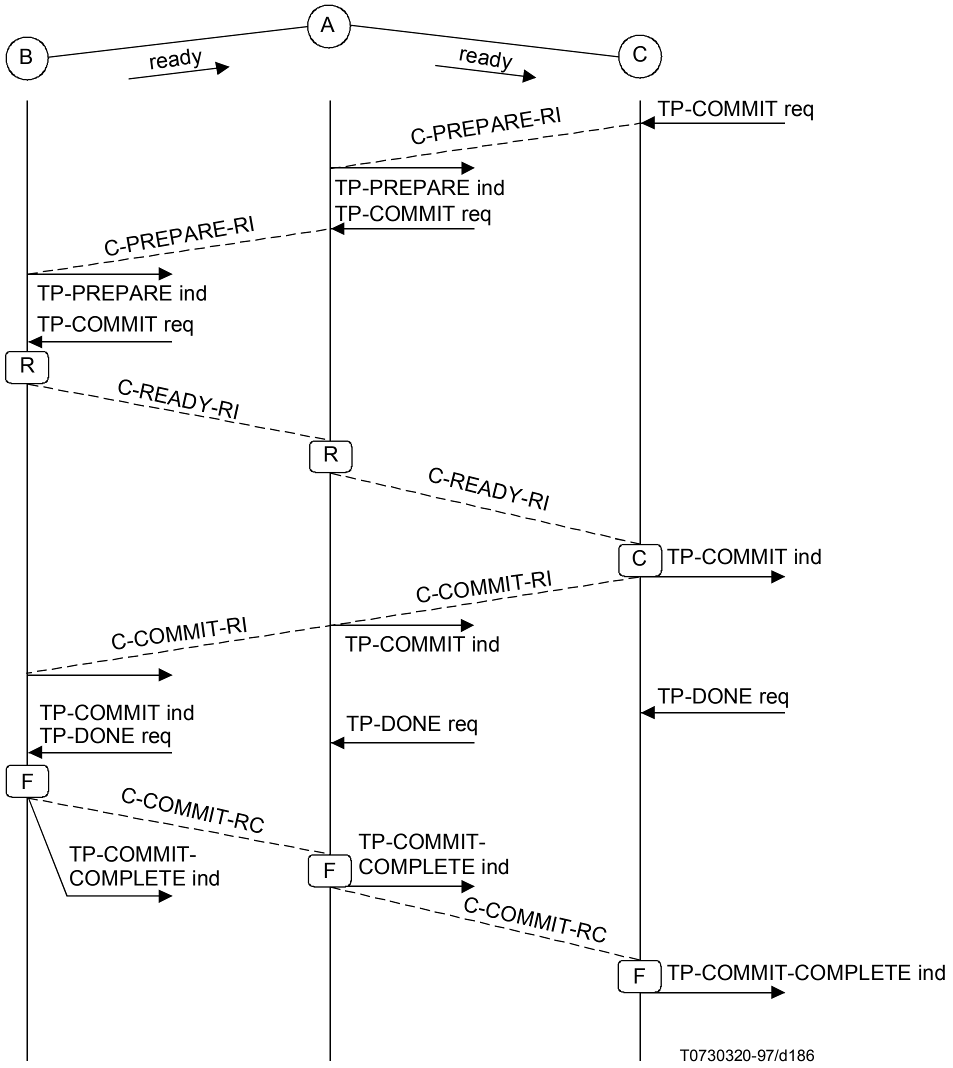


Figure C.180/X.862 – Alternate commit initiator

C.16.5 Alternate commit initiator with one-phase and read-only – No reporting

The scenario of Figure C.181 shows commitment being initiated from a non-root node. However the root issues a TP-ONE-PHASE request after receiving a TP-PREPARE indication from a subordinate, a further subordinate issues a TP-READ-ONLY request.

As a minimum the following functional units are selected:

- Dialogue A-C: Commit, Dynamic Commit, Unchained Transactions, Heuristic Report Suppression, Implicit Prepare, One-Phase Commit.

- Dialogue A-B: Commit, Unchained Transactions, Read-Only.

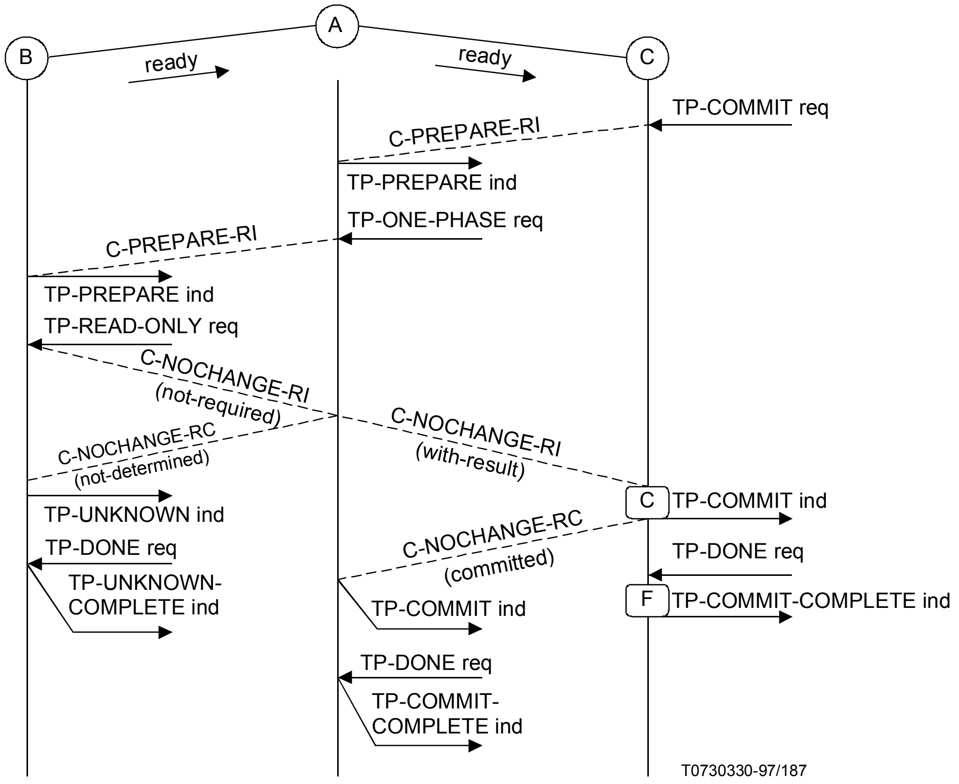


Figure C.181/X.862 – Alternate commit initiator with one-phase and read-only – No reporting

C.16.6 Alternate commit initiator with one-phase and read-only – With heuristic reporting

The scenario of Figure C.182 is a variation of the previous one, in which the Heuristic Report Suppression functional unit is not selected on the dialogue A-C. The TP-COMMIT-COMPLETE indication is delayed at A until all reports have been received. The scenario assumes that a TP-REPORT-RI with a Heuristic-Report parameter set to "none; otherwise a TP-HEURISTIC-REPORT indication would be issued to node A.

As a minimum the following functional units are selected:

– Dialogue A-C: Commit, Dynamic Commit, Unchained Transactions, Implicit Prepare, One-Phase Commit.

– Dialogue A-B: Commit, Unchained Transactions, Read-Only.

C.16.7 One-phase commit procedure with sending of C-PREPARE-RI followed by C-NOCHANGE-RI

In the scenario of Figure C.183, the root node (A) issues a TP-ONE-PHASE request; prepares are issued to all subordinates; when C issues a TP-READ-ONLY request, one-phase commit procedures can be used on the last dialogue. In the event of a one-phase/ready collision, the node C becomes the commit coordinator.

As a minimum the following functional units are selected:

– Dialogue A-C: Commit, Dynamic Commit, Unchained Transactions, Heuristic Report Suppression, Implicit Prepare, One-Phase Commit.

– Dialogue A-B: Commit, Unchained Transactions, Read-Only.

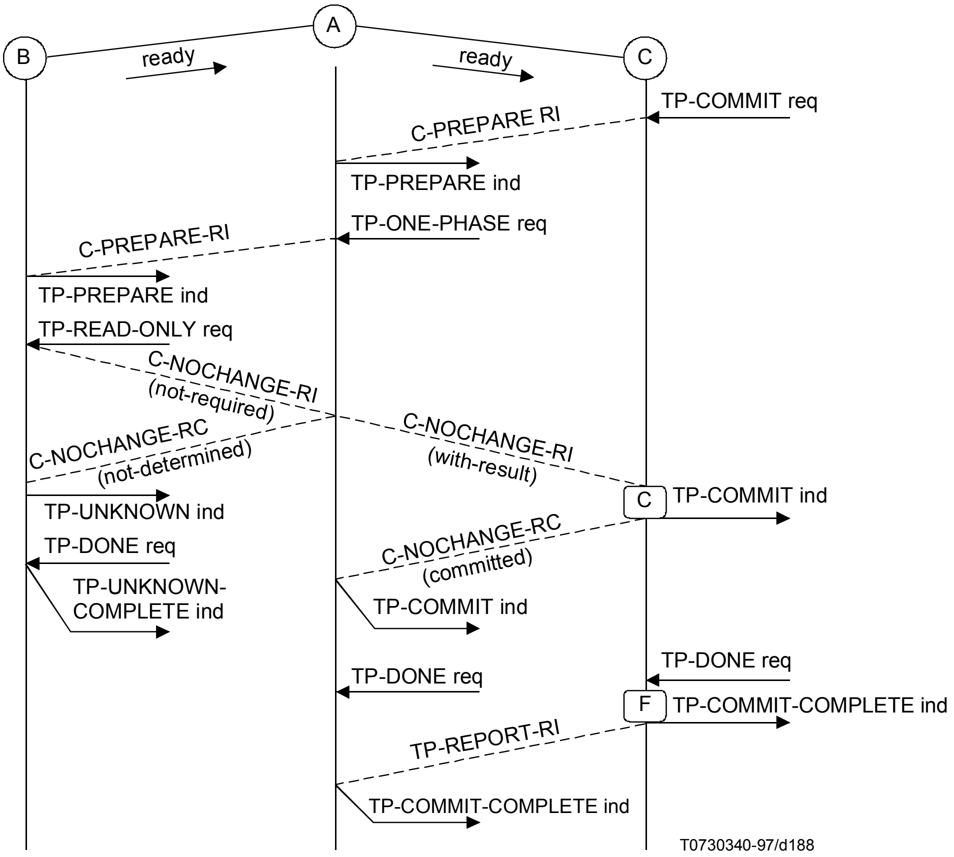


Figure C.182/X.862 – Alternate commit initiator with one-phase and read-only – With heuristic reporting

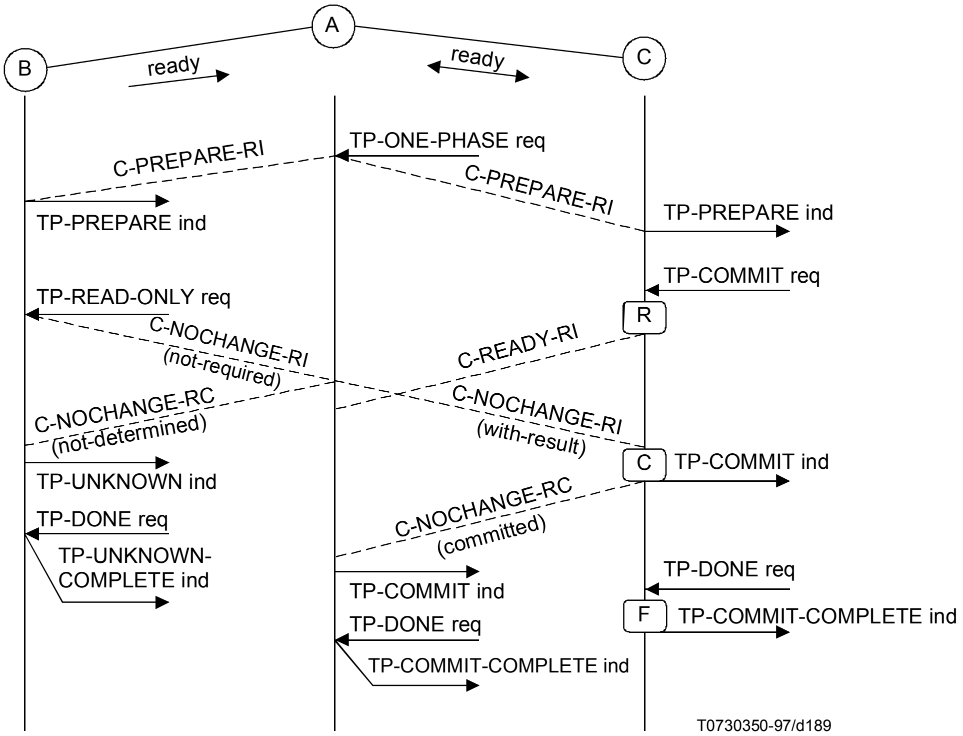


Figure C.183/X.862 – One-phase commit procedure with sending of C-PREPARE-RI followed by C-NOCHANGE-RI

C.16.8 One-phase above one-phase

Figure C.184 shows a scenario in which both root A and intermediate B use dynamic one-phase commitment, but the full commitment sequence occurs at C (possibly involving a further subordinate, not shown). The heuristic status is reported to A.

As a minimum the following functional units are selected on both dialogues: Commit, Dynamic Commit, Unchained Transactions, One-Phase Commit.

C writes a lazy log-forget.

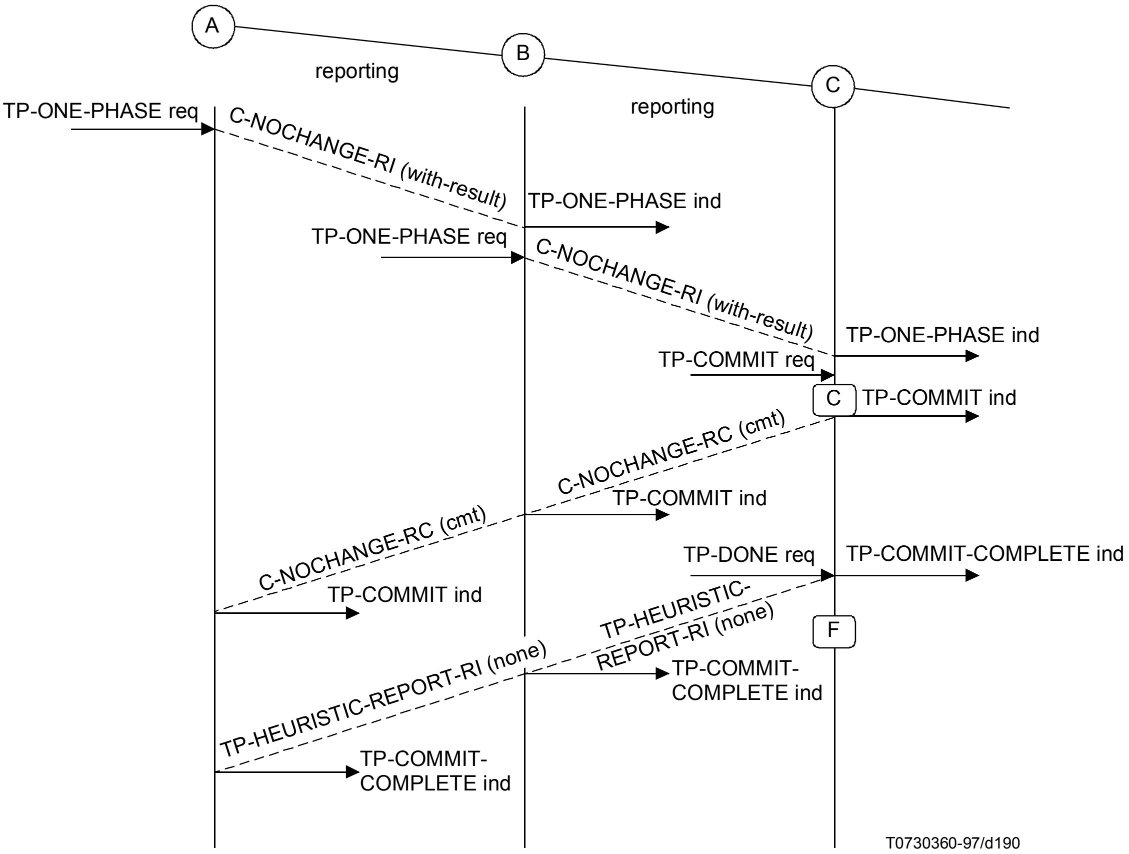


Figure C.184/X.862 – One-phase above one-phase

C.16.9 One-phase above one-phase – Unchained – No reporting

Figure C.185 is the same as the previous scenario, but no reporting on branch A-B.

As a minimum the following functional units are selected on both dialogues: Commit, Dynamic Commit, Unchained Transactions, One-Phase Commit; additionally the Heuristic Report Suppression functional unit is selected on dialogue A-B.

C.16.10 One-phase everywhere

Figure C.186 shows a transaction that changes no bound data at all. All three nodes issue TP-ONE-PHASE request. The leaf node makes the "one-phase decision".

As a minimum the following functional units are selected on both dialogues: Commit, Dynamic Commit, Unchained Transactions, One-Phase Commit; additionally the Heuristic Report Suppression functional unit is selected on dialogue A-B.

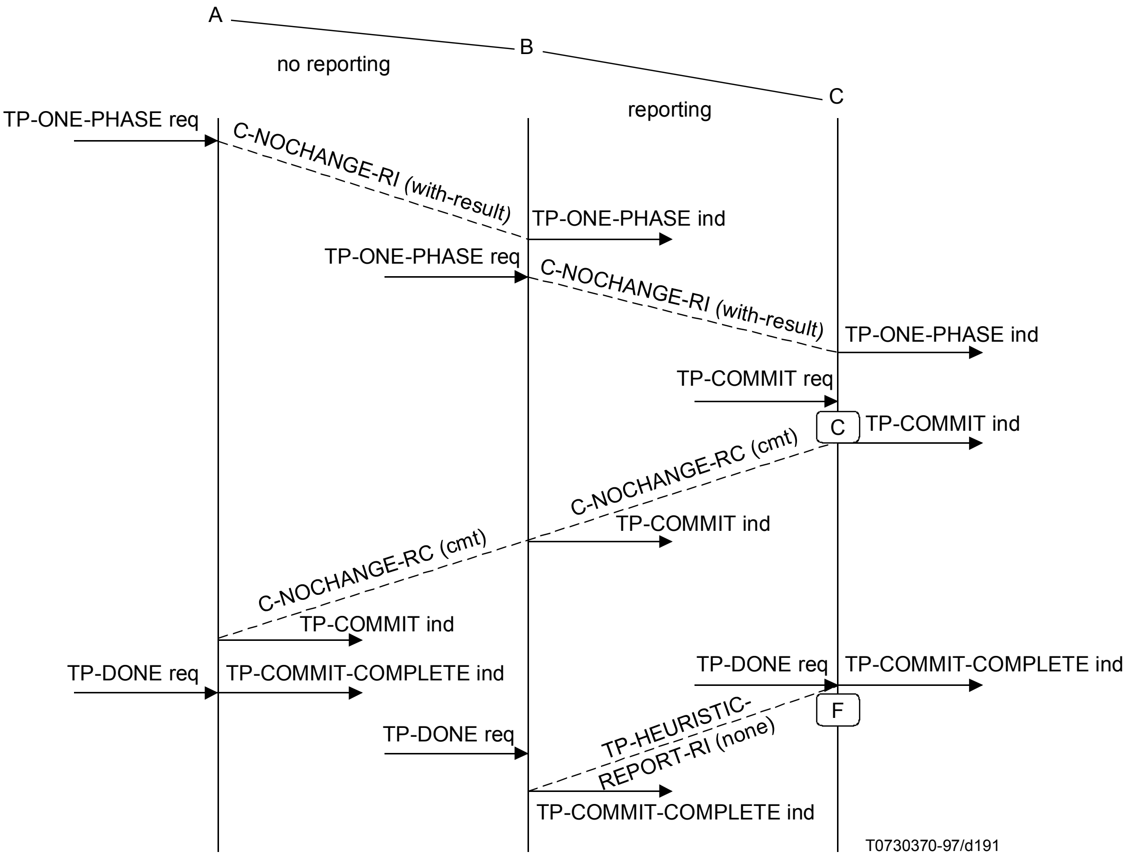


Figure C.185/X.862 – One-phase above one-phase – Unchained – No reporting

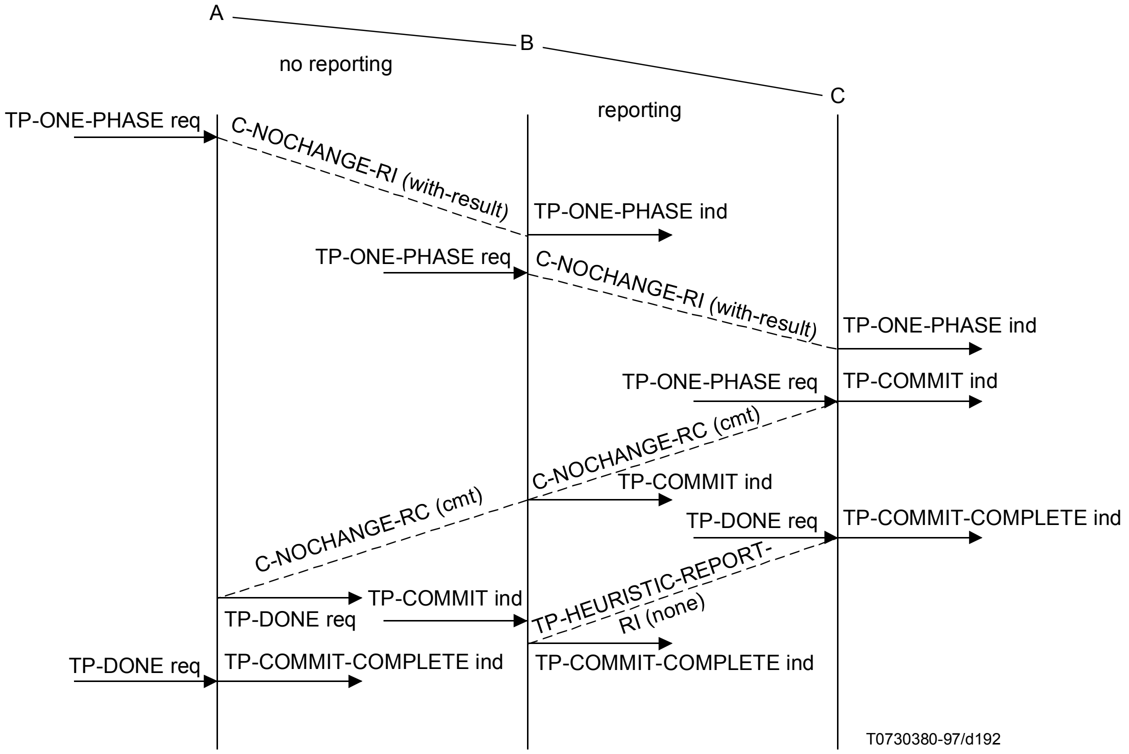


Figure C.186/X.862 – One-phase everywhere

C.16.11 Dynamic one-phase at root and intermediate, and read-only at leaf

Figure C.187 shows a "do-nothing" transaction, in which A and B issue TP-ONE-PHASE request and C issues TP-READ-ONLY request.

Functional units selected on both dialogues: Commit, Unchained Transactions, Dynamic Commit, Read-Only, One-phase; additionally the Heuristic Report Suppression functional unit is selected on dialogue A-B.

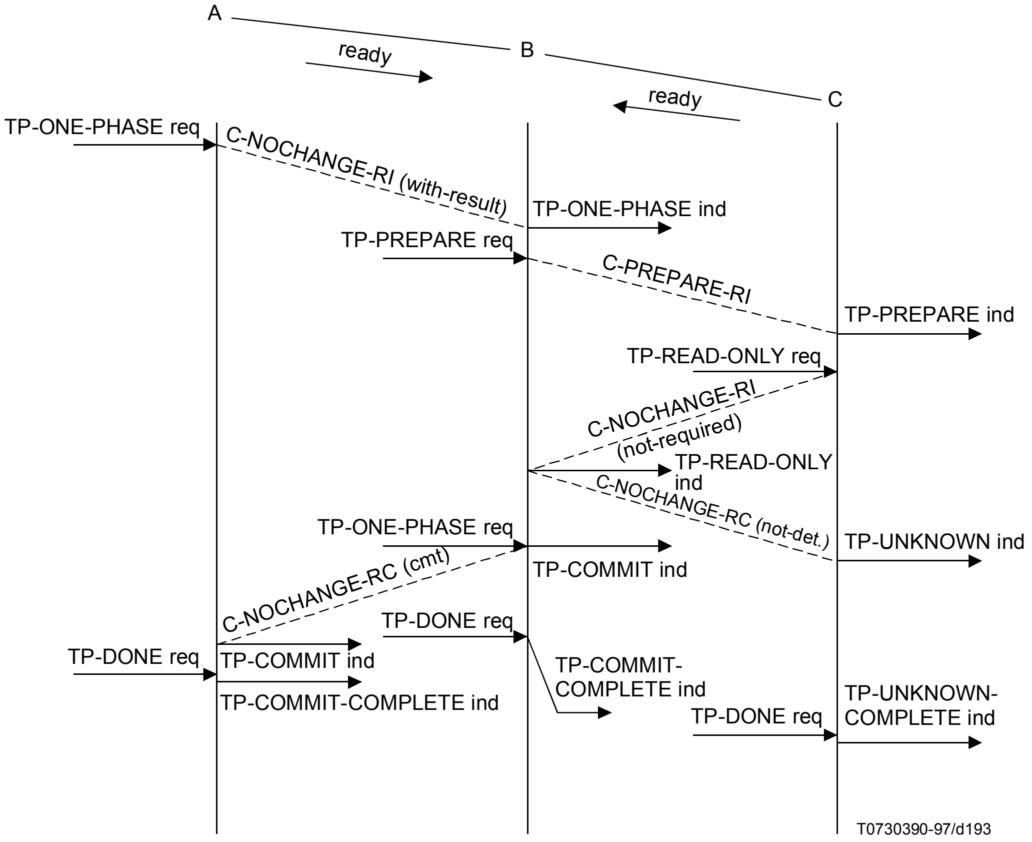


Figure C.187/X.862 – Dynamic one-phase at root and intermediate and read-only at leaf

C.16.12 Dynamic one-phase at root, and read-only at intermediate and leaf

Figure C.188 shows a "do-nothing" transaction, in which A issues TP-ONE-PHASE request and B and C issue TP-READ-ONLY request.

Functional units selected on both dialogues: Commit, Unchained Transactions, Dynamic Commit, Read-Only, One-phase, Heuristic Report Suppression.

C.16.13 One-phase and read only true collision

Figure C.189 shows a true collision between TP-ONE-PHASE and TP-READ-ONLY requests.

Functional units selected on both dialogues: Commit, Unchained Transactions, Dynamic Commit, Read-Only, One-phase. In addition, Implicit Prepare is selected on B-C.

C.17 Scenarios showing token movement during transaction termination

A number of scenarios follow showing the movement of the Session Layer synchronize-minor token. All show correct user behaviour. The superior is on the left. In each scenario, either the Shared Control and Implicit Prepare functional units are selected, or a TP-PREPARE request/indication was previously issued.

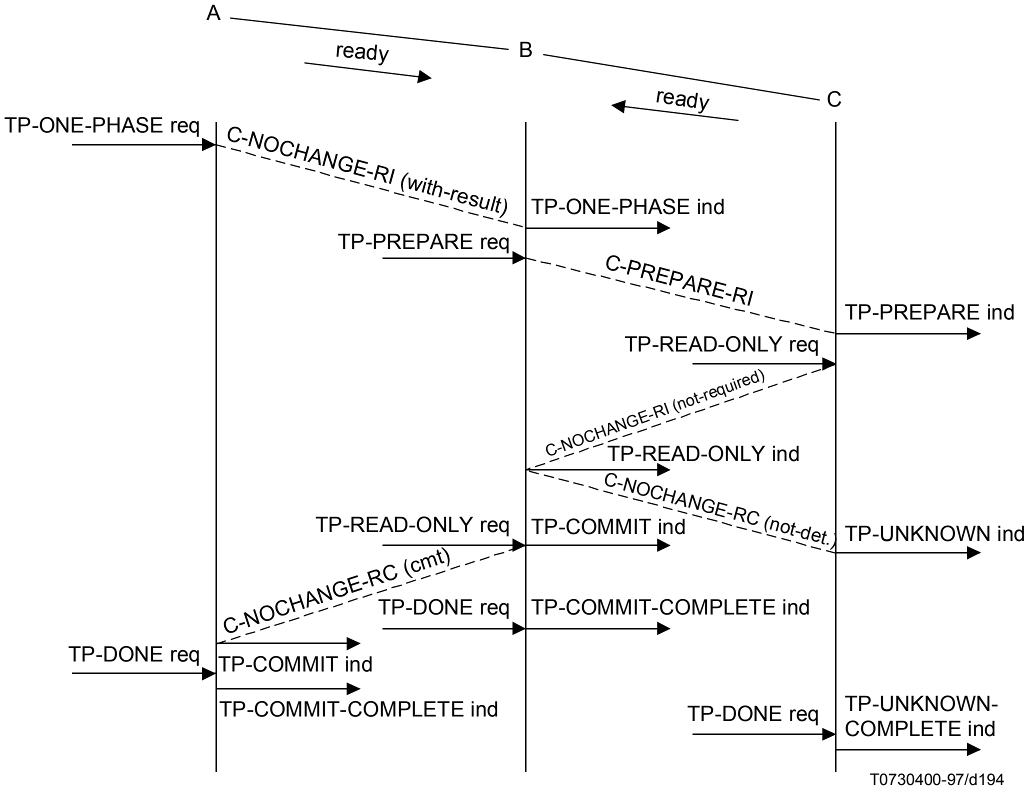


Figure C.188/X.862 – Dynamic one-phase at root, and read-only at intermediate and leaf

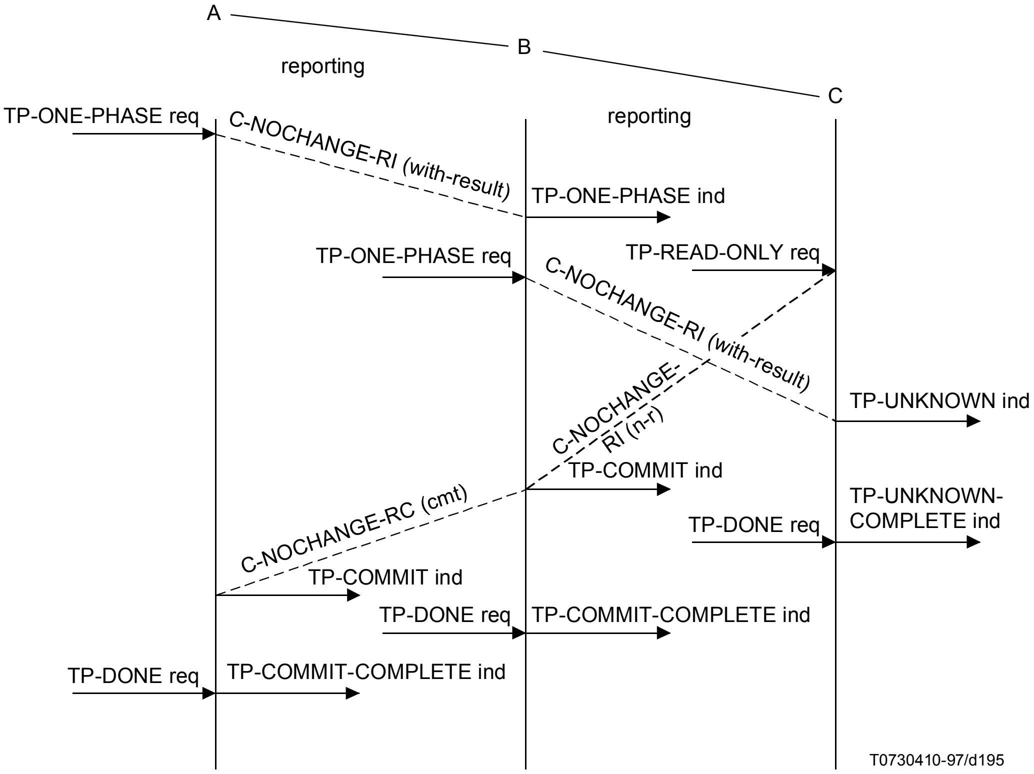
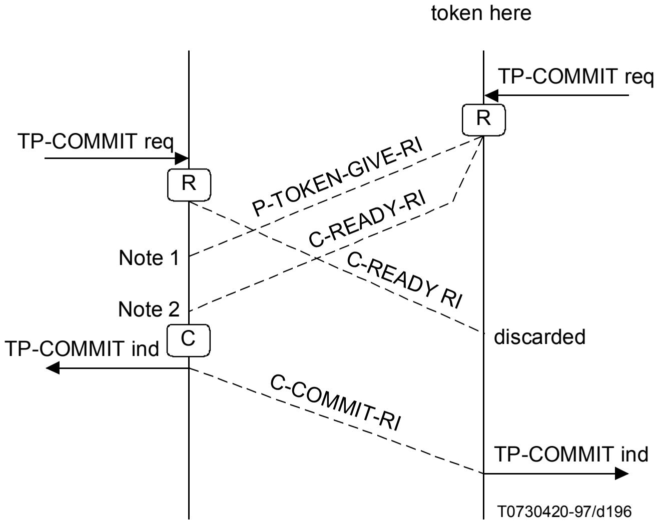


Figure C.189/X.862 – One-phase and read-only true collision

C.17.1 Ready/Ready collision – Superior becomes coordinator

Ready/ready collision: The node which receives the token having sent a signal of readiness becomes the commit coordinator.

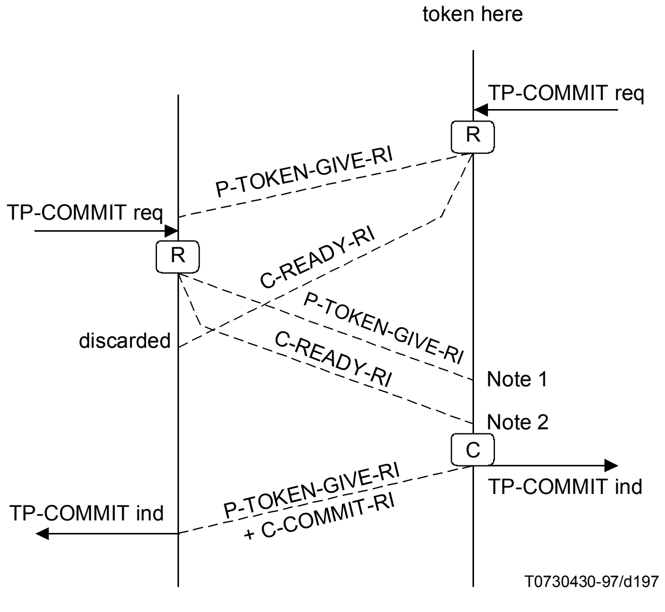


NOTE 1 – Arrival of the token indicates that ready or substitute will follow.

NOTE 2 – Node becomes the commit coordinator.

Figure C.190/X.862 – Ready/Ready collision – Superior becomes coordinator

C.17.2 Ready/Ready collision – Subordinate becomes coordinator



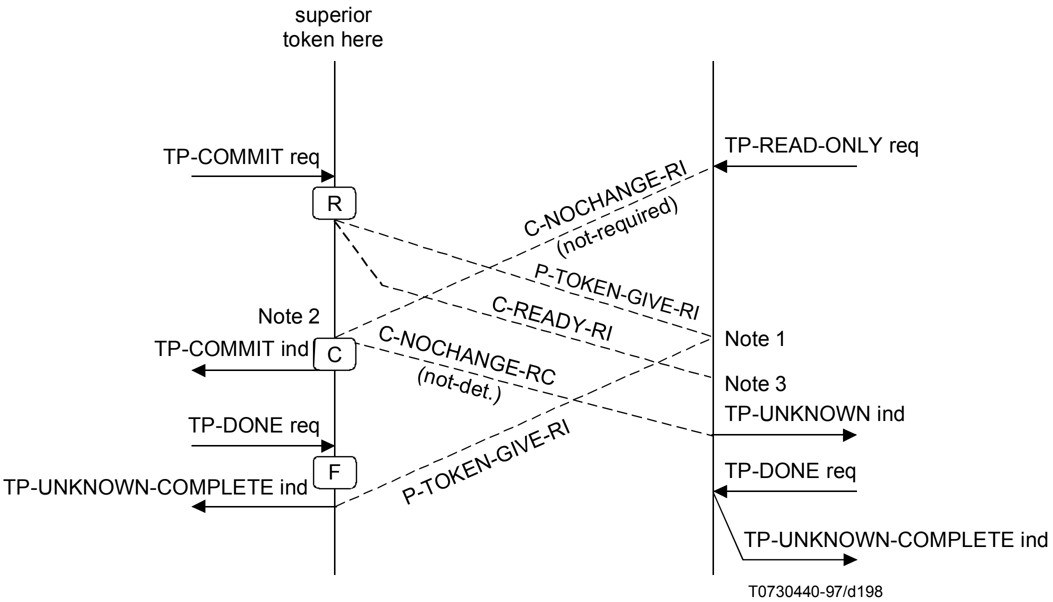
NOTE 1 – Arrival of the token indicates that ready or substitute will follow.

NOTE 2 – Node becomes the commit coordinator; the token is passed back ready for the next transaction.

Figure C.191/X.862 – Ready/Ready collision – Subordinate becomes coordinator

C.17.3 Ready/Read-Only collision – Superior becomes coordinator

Ready collision with one-phase or read-only: a node which receives a one phase or read-only signal having sent a signal of readiness becomes the commit coordinator; the tokens are returned to the superior ready for the beginning of the next transaction. The Unchained Transactions case is shown.



NOTE 1 – Arrival of the token indicates that ready or substitute will follow; however TPPM has already decided not to be commit coordinator and so token is returned to superior for next transaction even if unchained.

NOTE 2 – Node becomes the commit coordinator.

NOTE 3 – C-READY-RI is discarded (avoids unnecessary logging).

Figure C.192/X.862 – Ready/Read-Only collision – Superior becomes coordinator

C.17.4 Ready/One-phase collision – Subordinate becomes coordinator

See Figure C.193.

The case shown assumes no reporting; otherwise the C-NOCHANGE-RC will be delayed until the TP-DONE request is issued.

C.18 Recovery context handle on dialogue scenario

This subclause contains a scenario which illustrates the use of the RCH-on-dialogue functional unit.

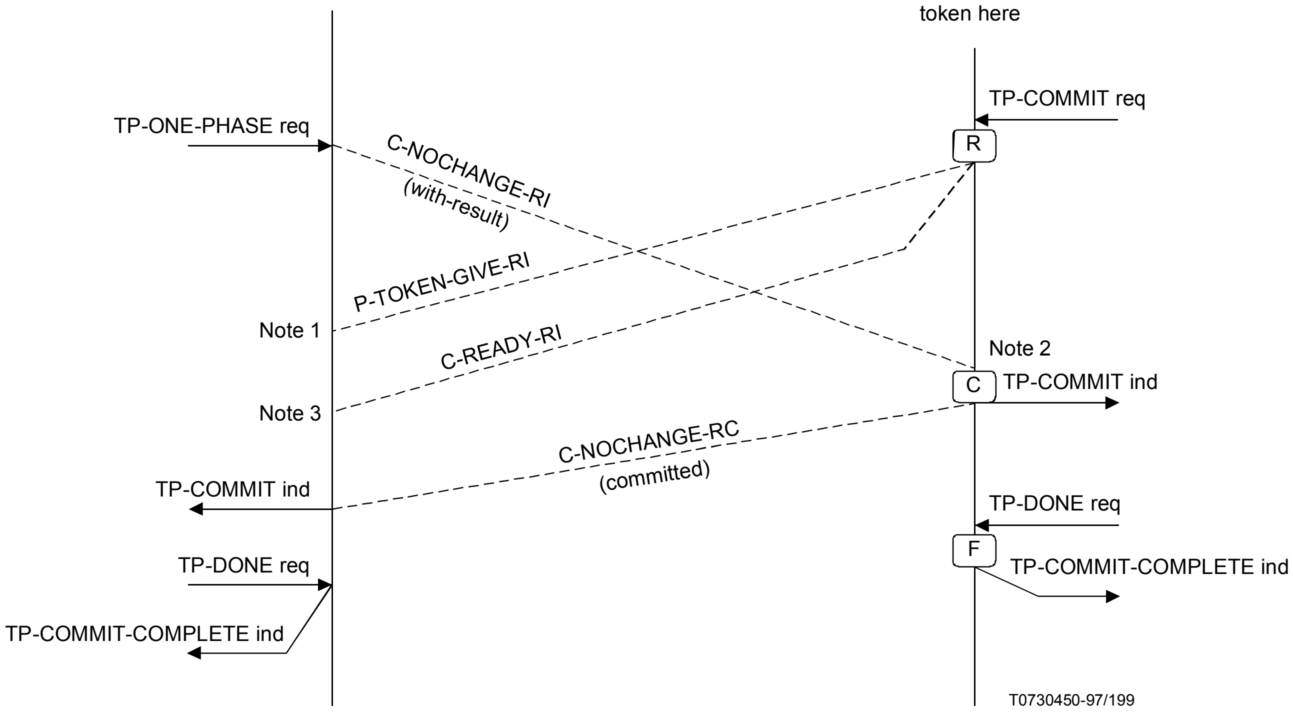
C.18.1 Late receipt of subordinate's RCH

There is only one scenario which is of significant interest. This shows the perhaps unusual case where the following combination of circumstances must all be true:

- A superior TPSUI begins a dialogue with Confirmation set to "negative" and either simultaneously or shortly after begins a transaction during which it initiates commitment before any response has been received from the subordinate.
- The Dynamic Commit functional unit is selected and Superior-may-send-ready shall be set to true (and Subordinate-may-send-ready shall be set to "false"?); therefore after a TP-COMMIT request, the superior TPPM would like to write a ready-log, but can not because the subordinate's RCH may change.

The circumstances to produce this are rather odd:

- It must be an unconfirmed TP-BEGIN-DIALOGUE request as otherwise the superior cannot invoke TP-COMMIT request until a response has been received.
- With Polarized Control, the subordinate can not make any sensible Service request because it does not have control, and therefore there is nothing that will provoke the generation of the TP-BEGIN-DIALOGUE-RC which may carry a new RCH (perhaps that is an API matter – somehow it happens).
- With Shared Control, there is still no sensible service request that can be issued at the subordinate.



NOTE 1 – Arrival of the token indicates that ready or substitute will follow; token remains with the superior.

NOTE 2 – Node becomes the commit coordinator.

NOTE 3 – C-READY-RI is discarded (avoids unnecessary logging).

Figure C.193/X.862 – Ready/One-phase collision – Subordinate becomes coordinator

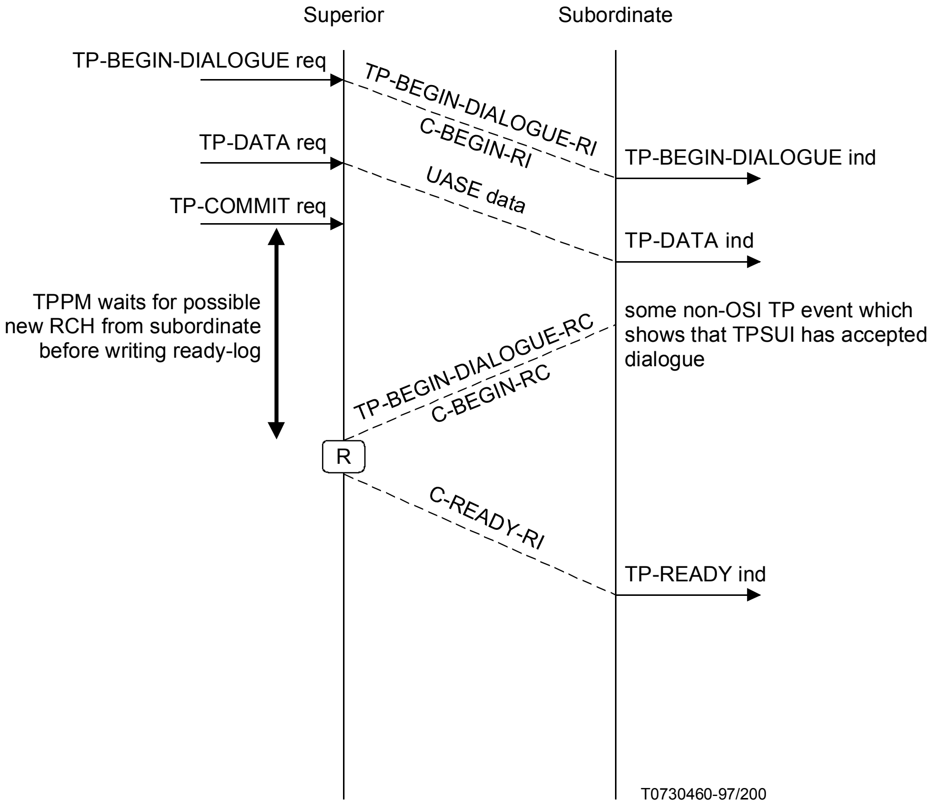


Figure C.194/X.862 – Late receipt of subordinate's RCH

Annex D

Summary of assigned object identifier values

This annex summarizes the OBJECT IDENTIFIER values assigned in this Recommendation.

tpASE1 APPLICATION-SERVICE-ELEMENT ::=

{joint-iso-ccitt transaction-processing(10) ase-id(0) tp-ase(1) version1(0)}

-- *may be used to identify the transaction processing ASE defined in*

-- *this Recommendation.*

{joint-iso-ccitt transaction-processing(10) modules (1)

apdus-abstract-syntax(1) version2(1)}

-- *used to reference the ASN.1 module defined in 12.1.*

id-as-tpase OBJECT IDENTIFIER ::=

{joint-iso-ccitt transaction-processing(10) abstract-syntax(2) tp-apdus(1)}

-- *may be used to reference the abstract syntax of the*

-- *transaction processing ASE APDUs*

Annex E

Recovery from destruction of atomic action data

E.1 Introduction

The correct operation of OSI TP provider supported transactions is dependent on the availability of a reliable recovery log as described in ITU-T Rec. X.860 | ISO/IEC 10026-1. The loss of some or all of the recovery log would mean that a system can not respond to requests to recover any or selected transactions; such failures could include total destruction of a real system with all its data, or partial failure such as loss of part or all of a recovery log.

If such a failure occurs during the later phases of a distributed transaction, then recovery of the transaction in the other nodes in the transaction tree would be unable to proceed; transactions will remain in the READY or DECIDED (COMMIT) state (depending on the role of the individual nodes) and recovery actions will either be rejected if a TP channel can not be established or receive a "retry-later" response.

Intermediate or leaf nodes which are affected because a superior node has suffered such a failure, may take heuristic decisions but that does not change the requirement to complete recovery and enter the "transaction forgotten" state. Such a node will have a log-ready record (and a log-heuristic record if a heuristic decision was taken) but using the mechanisms of OSI TP will not be able to determine the final outcome of the transaction (and whether any heuristic decision was correct or not).

Loss of atomic action data could occur as a result of several failures:

- partial loss of recovery log (for example due to media errors);
- complete loss of recovery log (for example due to a disc crash);
- destruction of system (for example by fire or flood);
- human operator error.

Because of the dependence on the recovery log for the correct operation of distributed transaction processing, many real systems will maintain multiple copies of the recovery log to minimize the chances of loss of all copies and may take other precautions against destruction of a system for example by dispersing the multiple copies across several geographical locations.

This annex is concerned with the case that in spite of measures to protect the recovery log contents, some loss has occurred.

E.2 Recovery actions

The following viewpoints are considered:

- a system which has been destroyed;
- a system which has lost part of its recovery log;
- a system which can not complete recovery because of lack of response from a further damaged system.

E.2.1 Actions after a system has been destroyed

This covers the case where a system cannot run at all or that it is sufficiently damaged that the TP Protocol machine cannot make any response to requests to establish a TP Channel for recovery. Such a system cannot take any useful action and is not considered further.

E.2.2 Actions of a node which has lost recovery records

Such a node must distinguish between recovery requests for transactions that it can be sure it has no record of and therefore must have completed as defined in this Recommendation, and recovery requests for transactions where the inaccessible recovery records might relate to the transaction in question. A node must not respond "unknown" or "done" for a transaction for which the recovery log record has been or may have been lost as this would imply that the transaction was terminated normally by either commit or rollback.

For example, a node may be certain that any recovery record related to a transaction is not affected by the loss because of the use of a recovery-context-handle which points to a set of recovery records which are unaffected.

Typically an alert would be raised for system management attention when loss of recovery records is detected; a system might also record requests for recovery to which it cannot safely respond to assist in the determination of the management actions that are required. Actions to determine the desired outcome of the transaction and to repair damage to applications and their bound data will be application dependent. Eventually the affected node must be repaired such that it can again function normally.

E.2.3 Actions of a node which cannot complete recovery

This concerns the case where a system cannot complete recovery of a transaction because of lack of response from a further damaged system which could either be a superior or subordinate node. It may be the case that a direct subordinate or superior is undamaged but that recovery is delayed because of a lack of response from a subordinate or superior node further down/up the transaction tree.

Such a node cannot directly determine whether recovery is delayed but will be possible eventually, or that recovery will never be possible. The node will observe either that a TP channel to the affected system cannot be established or that requests for recovery of the transaction are responded to by 'retry-later'.

In either case, typically after the expiration of a timer, the node may raise an alert for system management attention; this could prompt determination of the reason for lack of response and whether recovery will become possible or will remain impossible because of the conditions highlighted in this annex.

In the latter case, actions to determine the desired outcome of the transaction and to repair damage to applications and their bound data will be system or application dependent. When the desired outcome is determined, then an event must be caused to the TP implementation to remove recovery records related to the transaction such that it can finally be forgotten by the OSI TP Service Provider. In the terms of the main text and state tables for the OSI TP Protocol, this is equivalent to an internal event which instructs the TPPM to locally terminate the transaction. Such an action would be required in all affected nodes in the same transaction as there would be no means to propagate such an outcome throughout the transaction tree.

Annex F

TPPM transaction states

F.1 TPPM transaction states

At any time, a TPPM is either not part of a transaction tree or is in one of the following states:

– ACTIVE state;

– READY state;

– READ-ONLY state;

– EARLY-EXIT state;

– ONE-PHASE state;

- DECIDED (commit) state;
- DECIDED (rollback) state;
- DECIDED (unknown) state;
- DECIDED (commit-one-phase) state.

In the following subclauses, these states are defined by listing all legal state transitions (with reference numbers) and the conditions under which they occur. Figure F.1 gives a complete state transition diagram, and labels the arcs with the applicable state transition reference numbers.

The transaction state may change during the course of an action sequence according to the following rules. Should a node crash occur, the transaction state is derived from the existence of a log record.

F.1.1 State transitions of a root TPPM

1) A TPPM which is not part of a transaction tree becomes the root TPPM and enters the ACTIVE state when the TPPM receives:

   - a TP-BEGIN-DIALOGUE request with the Chained Transactions functional unit selected; or
   - a TP-BEGIN-DIALOGUE request with the Unchained Transactions functional unit selected and the Begin-Transaction parameter set to "true"; or
   - a TP-BEGIN-TRANSACTION request.

   This occurs in the "Initiating a transaction branch" procedure (see 11.5.9).

2) A root TPPM makes the transition from the ACTIVE state to the READY state when the *ready state may be entered* and the TPPM has written the log-ready record.

   This occurs in the "Entering READY state" procedure (see 11.5.6).

3) A root TPPM makes the transition from the ACTIVE state to the ONE-PHASE state when the *non-recovery states may be entered*.

   This occurs in the "Entering ONE-PHASE or READ-ONLY state" procedure (see 11.5.20).

4) A root TPPM makes the transition from the ACTIVE state to the DECIDED (commit) state when the *last ready has been received* and the log-commit record is written.

   This occurs in the "Making commitment decision" procedure (see 11.5.12).

5) A root TPPM makes the transition from the ACTIVE state to the DECIDED (rollback) state when the TPPM issues or receives a rollback-initiating service primitive.

   This occurs in the "Initiating rollback at TPPM" procedure (see 11.5.10).

6) A root TPPM in the ACTIVE state with only one subordinate ceases to be part of the transaction tree when the TPPM receives an AF-BEGIN-DIALOGUE (rejected) confirm or AF-BID (rejected) confirm on the dialogue with that subordinate, or receives an AF-ABORT (user/provider, dataRI) from its sole subordinate before receiving a TP-COMMIT request.

   This occurs in the AF-BEGIN-DIALOGUE (rejected, dataRI) confirm (see 11.3.5), SAF-ASSOCIATION-LOST indication" procedure (see 11.3.9), AF-BEGIN-DIALOGUE (rejected(user), rollbackRI) confirm (see 11.3.6), AF-ABORT (provider, dataRI) indication (see 11.3.20) procedures.

   A root TPPM in the ACTIVE state with only one subordinate also ceases to be part of the transaction tree when a TP-READ-ONLY indication or TP-EARLY-EXIT indication is issued on that dialogue and the dialogue is not chaining.

   This occurs in the AF-NOCHANGE indication or C-NOCHANGE indication procedure (see 11.3.62) and the AF-EARLY-EXIT indication procedure (see 11.3.64).

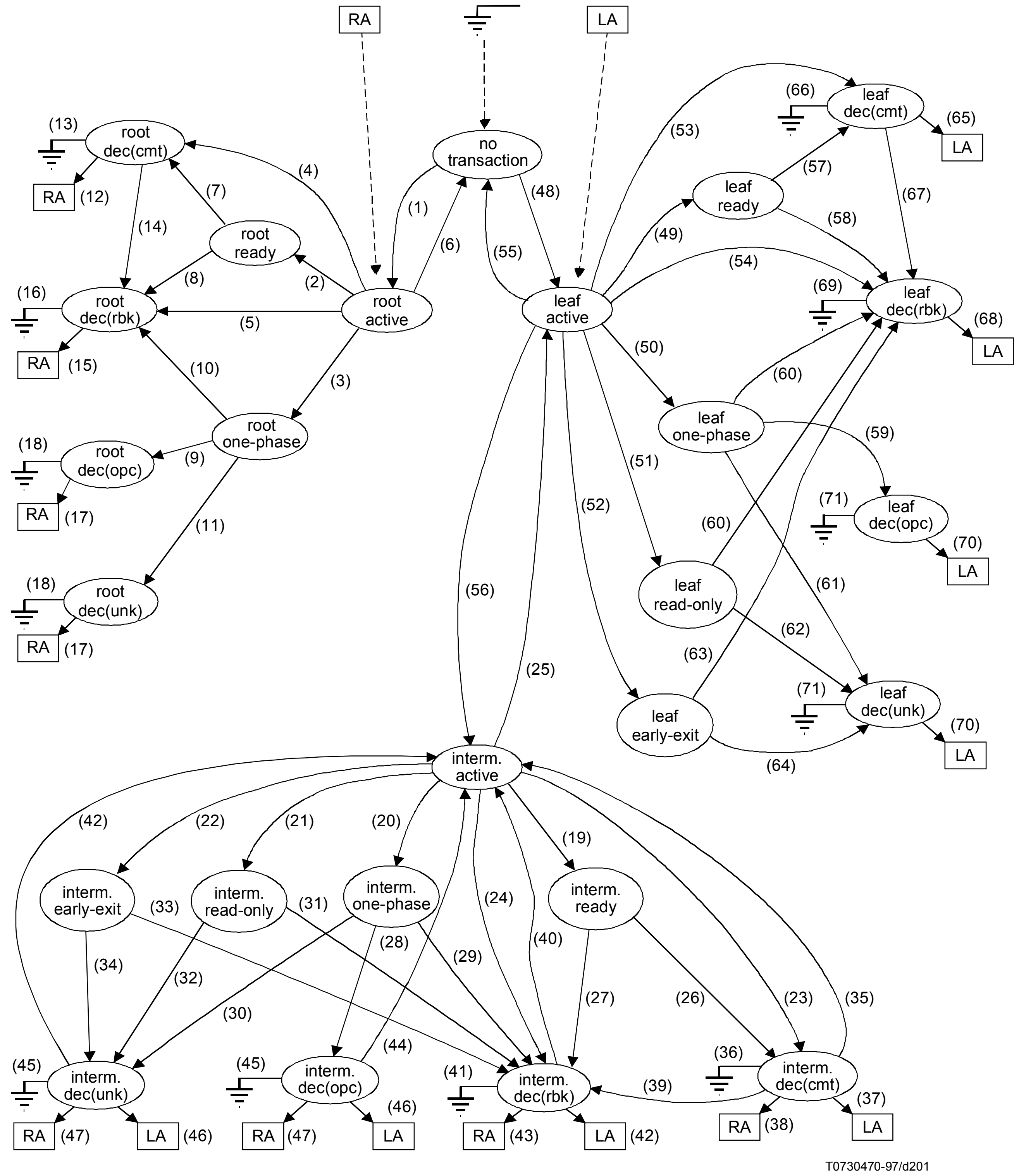


Figure F.1/X.862 – State transition diagram

7) A root TPPM makes the transition from the READY state to the DECIDED (commit) state when the TPPM receives *a commit indication* from the neighbour to whom a *ready signal* was sent.

   This occurs in the "Receiving commit order" procedure (see 11.5.14).

   The transition also occurs (following a ready-collision) when the *last ready was received* and the *token is owned* on the collision dialogue and the commit log-record is written.

   This occurs in the "Making commitment decision" procedure (see 11.5.12).

8) A root TPPM makes the transition from the READY state to the DECIDED (rollback) state when it receives a *rollback indication* from the neighbour to whom it sent a *ready signal.*

   The transition also occurs (following a ready-collision) when the *last ready was received* and the *token is owned* on the collision dialogue but the commit log-record cannot be written.

   The transition occurs in the "Initiating rollback at TPPM" procedure (see 11.5.10).

9) A root TPPM makes the transition from the ONE-PHASE state to the DECIDED (commit-one-phase) state when a C-NOCHANGE confirm of AF-ABORT (user, nochangeRC) indication with outcome parameter "commit" or "no-change" is received.

   This occurs in the "C-NOCHANGE confirm or AF-ABORT (user, nochangeRC) indication" procedure (see 11.3.68).

10) A root TPPM makes the transition from the ONE-PHASE state to the DECIDED (rollback) state when the TPPM receives a *rollback indication* from the neighbour to whom it sent the C-NOCHANGE or AF-NOCHANGE request.

    This occurs in the "Initiating rollback at TPPM" procedure (see 11.5.10).

11) A root TPPM makes the transition from the ONE-PHASE state to the DECIDED (unknown) state when a C-NOCHANGE confirm of AF-ABORT (user, nochangeRC) indication with outcome parameter "not-determined" is received or when there is a failure of the dialogue on which the C-NOCHANGE request or AF-NOCHANGE request was issued.

    This occurs in the "C-NOCHANGE confirm or AF-ABORT (user, nochangeRC) indication" procedure (see 11.3.68) and in the "Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm" procedure (see 11.3.21).

12) A root TPPM makes the transition from the DECIDED (commit) state to the ACTIVE state when the TPPM issues a TP-COMMIT-COMPLETE indication provided at least one subordinate *dialogue is chaining*.

    This occurs in the "Completing commitment" procedure (see 11.5.3).

13) A root TPPM that is in the DECIDED (commit) state ceases to be part of the transaction tree when the TPPM issues a TP-COMMIT-COMPLETE indication provided that no subordinate *dialogue is chaining*.

    This occurs in the "Completing commitment" procedure (see 11.5.3).

14) A root TPPM makes the transition from the DECIDED (commit) state to the DECIDED (rollback) state when the TPPM issues a TP-COMMIT-COMPLETE indication followed by a TP-ROLLBACK indication.

    This occurs in the "Completing commitment" procedure (see 11.5.3).

15) A root TPPM makes the transition from the DECIDED (rollback) state to the ACTIVE state when the TPPM issues a TP-ROLLBACK-COMPLETE indication provided at least one subordinate *dialogue is chaining*.

    This occurs in the "Initiating transaction after rollback" procedure (see 11.5.11).

16) A root TPPM that is in the DECIDED (rollback) state ceases to be part of the transaction tree when the TPPM issues a TP-ROLLBACK-COMPLETE indication provided that no subordinate *dialogue is chaining*.

    This occurs in the "Initiating transaction after rollback" procedure (see 11.5.11).

17) A root TPPM makes the transition from the DECIDED (commit-one-phase) state [or DECIDED (unknown) state] to the ACTIVE state when the TPPM issues a TP-COMMIT-COMPLETE indication (or TP-UNKNOWN-COMPLETE, respectively) provided at least one subordinate *dialogue is chaining*.

    This occurs in the "Completing one-phase and read-only" procedure (see 11.5.5).

18) A root TPPM that is in the DECIDED (commit-one-phase) state [or DECIDED (unknown) state] ceases to be part of the transaction tree when the TPPM issues a TP-COMMIT-COMPLETE indication (or TP-UNKNOWN-COMPLETE indication) provided that no subordinate *dialogue is chaining*.

    This occurs in the "Completing one-phase and read-only" procedure (see 11.5.5).

F.1.2 State transitions of an intermediate TPPM

19) An intermediate TPPM makes the transition from the ACTIVE state to the READY state when the *ready state may be entered* and the TPPM has written the log-ready record.

    This occurs in the "Entering READY state" procedure (see 11.5.6).

20) An intermediate TPPM makes the transition from the ACTIVE state to the ONE-PHASE state when the *non-recovery states may be entered* and a TP-ONE-PHASE request was received or a TP-READ-ONLY request was received but at least one C-NOCHANGE (result-requested) indication was received.

    This occurs in the "Entering ONE-PHASE or READ-ONLY state" procedure (see 11.5.20).

21) An intermediate TPPM makes the transition from the ACTIVE state to the READ-ONLY state when the *non-recovery states may be entered* and a TP-READ-ONLY request was received and no C-NOCHANGE (result-requested) indication was received.

    This occurs in the "Entering ONE-PHASE or READ-ONLY state" procedure (see 11.5.20).

22) An intermediate TPPM makes the transition from the ACTIVE state to the EARLY-EXIT state when a TP-EARLY-EXIT request is received and there is no TPPM bound data.

    This occurs in the "TP-EARLY-EXIT request" procedure (see 11.3.63).

23) An intermediate TPPM makes the transition from the ACTIVE state to the DECIDED (commit) state when the *last ready has been received* and the log-commit record is written.

    This occurs in the "Making commitment decision" procedure (see 11.5.12).

24) An intermediate TPPM makes the transition from the ACTIVE state to the DECIDED (rollback) state when the TPPM issues or receives a rollback-initiating service primitive.

    This occurs in the "Initiating rollback at TPPM" procedure (see 11.5.10).

25) An intermediate TPPM in the ACTIVE state with only one subordinate becomes a leaf in the ACTIVE state when a TP-READ-ONLY indication or TP-EARLY-EXIT indication is issued on the sole subordinate dialogue and the dialogue is not chaining.

    This occurs in the AF-NOCHANGE indication or C-NOCHANGE indication procedure (see 11.3.62) and the AF-EARLY-EXIT indication procedure (see 11.3.64).

26) An intermediate TPPM makes the transition from the READY state to the DECIDED (commit) state when the TPPM receives *a commit indication* from the neighbour to whom a *ready signal* was sent.

    This occurs in the "Receiving commit order" procedure (see 11.5.14).

    The transition also occurs (following a ready-collision) when the *last ready was received* and the *token is owned* on the collision dialogue and the commit log-record is written.

    This occurs in the "Making commitment decision" procedure (see 11.5.12).

27) An intermediate TPPM makes the transition from the READY state to the DECIDED (rollback) state when the TPPM receives *a rollback indication* from the neighbour to whom a *ready signal was sent*.

    The transition also occurs (following a ready-collision) when the *last ready was received* and the *token is owned* on the collision dialogue but the commit log-record cannot be written.

    The transition occurs in the "Initiating rollback at TPPM" procedure (see 11.5.10).

28) An intermediate TPPM makes the transition from the ONE-PHASE state to the DECIDED (commit-one-phase) state when a C-NOCHANGE confirm of AF-ABORT (user, nochangeRC) indication with outcome parameter "commit" or "no-change" is received.

    This occurs in the "C-NOCHANGE confirm or AF-ABORT (user, nochangeRC) indication" procedure (see 11.3.68).

29) An intermediate TPPM makes the transition from the ONE-PHASE state to the DECIDED (rollback) state when the TPPM receives a *rollback indication* from the neighbour to whom it sent the C-NOCHANGE or AF-NOCHANGE request.

This occurs in the "Initiating rollback at TPPM" procedure (see 11.5.10).

30) An intermediate TPPM makes the transition from the ONE-PHASE state to the DECIDED (unknown) state when a C-NOCHANGE confirm of AF-ABORT (user, nochangeRC) indication with outcome parameter "not-determined" is received or when there is a failure of the dialogue on which the C-NOCHANGE request or AF-NOCHANGE request was issued.

This occurs in the "C-NOCHANGE confirm or AF-ABORT (user, nochangeRC) indication" procedure (see 11.3.68) and in the "Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm" procedure (see 11.3.21).

31) An intermediate TPPM makes the transition from the READ-ONLY state to the DECIDED (rollback) state when the TPPM receives a *rollback indication* from the superior.

This occurs in the "Initiating rollback at TPPM" procedure (see 11.5.10).

32) An intermediate TPPM makes the transition from the READ-ONLY state to the DECIDED (unknown) state when a C-NOCHANGE confirm or AF-ABORT (user, nochangeRC) indication is received or when a C-BEGIN indication is received or when there is a failure of the superior dialogue.

These occur in the "C-NOCHANGE confirm or AF-ABORT (user, nochangeRC) indication" procedure (see 11.3.68), in the "C-BEGIN indication or AF-BEGIN-TRANSACTION indication" procedure (see 11.3.37) and in the "Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm" procedure (see 11.3.21).

33) An intermediate TPPM makes the transition from the EARLY-EXIT state to the DECIDED (rollback) state when the TPPM receives a *rollback indication* or C-ROLLBACK confirm from the superior.

This occurs in the "Initiating rollback at TPPM" procedure (see 11.5.10).

34) An intermediate TPPM makes the transition from the EARLY-EXIT state to the DECIDED (unknown) state when an AF-EARLY-EXIT confirm is received or when there is a failure of the superior dialogue.

These occur in the "AF-EARLY-EXIT confirm" procedure (see 11.3.65) and in the "Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm" procedure (see 11.3.21).

35) An intermediate TPPM makes the transition from the DECIDED (commit) state to the ACTIVE state when the TPPM issues TP-COMMIT-COMPLETE indication provided that the superior and at least one subordinate *dialogue is chaining*.

This occurs in the "Completing commitment" procedure (see 11.5.3).

36) An intermediate TPPM that is in the DECIDED (commit) state ceases to be part of the transaction tree when the TPPM issues TP-COMMIT-COMPLETE indication provided that no *dialogue is chaining*.

This occurs in the "Completing commitment" procedure (see 11.5.3).

37) An intermediate TPPM in the DECIDED (commit) state becomes a leaf TPPM in the ACTIVE state when the TPPM issues TP-COMMIT-COMPLETE indication provided that the superior *dialogue is chaining* and no subordinate *dialogue is chaining*.

This occurs in the "Completing commitment" procedure (see 11.5.3).

38) An intermediate TPPM in the DECIDED (commit) state becomes a root TPPM in the ACTIVE state when the TPPM issues TP-COMMIT-COMPLETE indication provided that the superior *dialogue is not chaining* and at least one subordinate *dialogue is chaining*.

This occurs in the "Completing commitment" procedure (see 11.5.3).

39) An intermediate TPPM makes the transition from the DECIDED (commit) state to the DECIDED (rollback) state when the TPPM issues a TP-COMMIT-COMPLETE indication followed by a TP-ROLLBACK indication.

This occurs in the "Completing commitment" procedure (see 11.5.3).

40) An intermediate TPPM makes the transition from the DECIDED (rollback) state to the ACTIVE state when the TPPM issues a TP-ROLLBACK-COMPLETE indication provided that the superior and at least one subordinate *dialogue is chaining*.

This occurs in the "Initiating transaction after rollback" procedure (see 11.5.11).

41) An intermediate TPPM that is in the DECIDED (rollback) state ceases to be part of the transaction tree when the TPPM issues a TP-ROLLBACK-COMPLETE indication provided that no *dialogue is chaining*.

This occurs in the "Initiating transaction after rollback" procedure (see 11.5.11).

42) An intermediate TPPM in the DECIDED (rollback) state becomes a leaf TPPM in the ACTIVE state when the TPPM issues a TP-ROLLBACK-COMPLETE indication provided that the superior *dialogue is chaining* and no subordinate *dialogue is chaining*.

This occurs in the "Initiating transaction after rollback" procedure (see 11.5.11).

43) An intermediate TPPM in the DECIDED (rollback) state becomes a root TPPM in the ACTIVE state when the TPPM issues a TP-ROLLBACK-COMPLETE indication provided that the superior *dialogue is not chaining* and at least one subordinate *dialogue is chaining*.

This occurs in the "Initiating transaction after rollback" procedure (see 11.5.11).

44) An intermediate TPPM makes the transition from the DECIDED (commit-one-phase) state [or DECIDED (unknown) state] to the ACTIVE state when the TPPM issues a TP-COMMIT-COMPLETE indication (or TP-UNKNOWN-COMPLETE indication, respectively) provided that the superior and at least one subordinate *dialogue is chaining*.

This occurs in the "Completing one-phase and read-only" procedure (see 11.5.5).

45) An intermediate TPPM that is in the DECIDED (commit-one-phase) state [or DECIDED (unknown) state] ceases to be part of the transaction tree when the TPPM issues a TP-COMMIT-COMPLETE indication (or TP-UNKNOWN-COMPLETE indication, respectively) provided that no *dialogue is chaining*.

This occurs in the "Completing one-phase and read-only" procedure (see 11.5.5).

46) An intermediate TPPM in the DECIDED (commit-one-phase) state [or DECIDED (unknown) state] becomes a leaf TPPM in the ACTIVE state when the TPPM issues a TP-COMMIT-COMPLETE indication (or TP-UNKNOWN-COMPLETE indication, respectively) provided that the superior *dialogue is chaining* and no subordinate *dialogue is chaining*.

This occurs in the "Completing one-phase and read-only" procedure (see 11.5.5).

47) An intermediate TPPM in the DECIDED (commit-one-phase) state [or DECIDED (unknown) state] becomes a root TPPM in the ACTIVE state when the TPPM issues a TP-COMMIT-COMPLETE indication (or TP-UNKNOWN-COMPLETE indication, respectively) provided that the superior *dialogue is not chaining* and at least one subordinate *dialogue is chaining*.

This occurs in the "Completing one-phase and read-only" procedure (see 11.5.5).

F.1.3 State transitions of a leaf TPPM

48) A TPPM which is not part of a transaction tree becomes a leaf TPPM and enters the ACTIVE state when the TPPM receives a C-BEGIN indication or AF-BEGIN-TRANSACTION indication.

This occurs in the "C-BEGIN indication or AF-BEGIN-TRANSACTION indication" procedure (see 11.3.37).

49) A leaf TPPM makes the transition from the ACTIVE state to the READY state when *ready state may be entered* and the TPPM has written the log-ready record.

This occurs in the "Entering READY state" procedure (see 11.5.6).

50) A leaf TPPM makes the transition from the ACTIVE state to the ONE-PHASE state when the *non-recovery states may be entered* and a TP-ONE-PHASE request was received.

This occurs in the "Entering ONE-PHASE or READ-ONLY state" procedure (see 11.5.20).

51) A leaf TPPM makes the transition from the ACTIVE state to the READ-ONLY state when the *non-recovery states may be entered* and a TP-READ-ONLY request was received.

This occurs in the "Entering ONE-PHASE or READ-ONLY state" procedure (see 11.5.20).

52) A leaf TPPM makes the transition from the ACTIVE state to the EARLY-EXIT state when a TP-EARLY-EXIT request is received and there is no TPPM bound data.

This occurs in the "TP-EARLY-EXIT request" procedure (see 11.3.63).

53) A leaf TPPM makes the transition from the ACTIVE state to the DECIDED (commit) state when the *last ready has been received* and the log-commit record is written.

This occurs in the "Making commitment decision" procedure (see 11.5.12).

54) A leaf TPPM makes the transition from the ACTIVE state to the DECIDED (rollback) state when the TPPM issues or receives a rollback-initiating service primitive.

This occurs in the "Initiating rollback at TPPM" procedure (see 11.5.10).

55) A leaf TPPM which is in the ACTIVE state ceases to be part of the transaction tree when the TPPM receives a TP-BEGIN-DIALOGUE (rejected) response, or when there is a dialogue establishment indication outstanding and an AF-ABORT (user/provider, rollbackRI) indication is received.

This occurs in the TP-BEGIN-DIALOGUE response" procedure (see 11.3.3) and in the AF-ABORT (user/provider, rollbackRI) indication or AF-ABORT-AND-REPORT (rollbackRI) indication procedure (see 11.3.56).

56) A leaf TPPM which is in the ACTIVE state becomes an intermediate TPPM in the ACTIVE state when the TPPM:

– issues an AF-BEGIN-DIALOGUE request with selection of the Chained Transactions functional unit; or

– issues an AF-BEGIN-DIALOGUE request with selection of the Unchained Transactions functional unit and the Begin-Transaction parameter set to "commitment"; or

– receives a TP-BEGIN-TRANSACTION request.

This occurs in the "Initiating a transaction branch" procedure (see 11.5.9).

57) A leaf TPPM makes the transition from the READY state to the DECIDED (commit) state when the TPPM receives a *commit indication* from the superior.

This occurs in the "Receiving commit order" procedure (see 11.5.14).

The transition also occurs (following a ready-collision) when the *last ready was received* and the *token is owned* and the commit log-record is written.

This occurs in the "Making commitment decision" procedure (see 11.5.12).

58) A leaf TPPM makes the transition from the READY state to the DECIDED (rollback) state when the TPPM receives a *rollback indication* from its superior.

The transition also occurs (following a ready-collision) when the *last ready was received* and the *token is owned* on the collision dialogue but the commit log-record cannot be written.

The transition occurs in the "Initiating rollback at TPPM" procedure (see 11.5.10).

59) A leaf TPPM makes the transition from the ONE-PHASE state to the DECIDED (commit-one-phase) state when a C-NOCHANGE confirm of AF-ABORT (user, nochangeRC) indication with outcome parameter "commit" or "no-change" is received.

This occurs in the "C-NOCHANGE confirm or AF-ABORT (user, nochangeRC) indication" procedure (see 11.3.68).

60) A leaf TPPM makes the transition from the ONE-PHASE state or READ-ONLY state to the DECIDED (rollback) state when the TPPM receives a *rollback indication* from the superior.

This occurs in the "Initiating rollback at TPPM" procedure (see 11.5.10).

61) A leaf TPPM makes the transition from the ONE-PHASE state to the DECIDED (unknown) state when a C-NOCHANGE confirm or AF-ABORT (user, nochangeRC) indication with outcome parameter "not-determined" is received or when there is a failure of the superior dialogue.

This occurs in the "C-NOCHANGE confirm or AF-ABORT (user, nochangeRC) indication" procedure (see 11.3.68) and in the "Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm" procedure (see 11.3.21).

62) A leaf TPPM makes the transition from the READ-ONLY state to the DECIDED (unknown) state when a C-NOCHANGE confirm or AF-ABORT (user, nochangeRC) indication is received or when a C-BEGIN indication or AF-BEGIN-TRANSACTION indication is received or when there is a failure of the superior dialogue.

These occur in the "C-NOCHANGE confirm or AF-ABORT (user, nochangeRC) indication" procedure (see 11.3.68), in the "C-BEGIN indication or AF-BEGIN-TRANSACTION indication" procedure (see 11.3.37) and in the "Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm" procedure (see 11.3.21).

63) A leaf TPPM makes the transition from the EARLY-EXIT state to the DECIDED (rollback) state when the TPPM receives a *rollback indication* or C-ROLLBACK confirm from the superior.

This occurs in the "Initiating rollback at TPPM" procedure (see 11.5.10).

64) A leaf TPPM makes the transition from the EARLY-EXIT state to the DECIDED (unknown) state when an AF-EARLY-EXIT confirm is received or when there is a failure of the superior dialogue.

These occur in the "AF-EARLY-EXIT confirm" procedure (see 11.3.65) and in the "Protocol error, internal error, A[-P]-ABORT indication, AF-ABORT (provider, abortRI) indication, A-ABORT request, A-RELEASE (Result = affirmative) response, or A-RELEASE (Result = affirmative) confirm" procedure (see 11.3.21).

65) A leaf TPPM makes the transition from the DECIDED (commit) state to the ACTIVE state when the TPPM issues a TP-COMMIT-COMPLETE indication provided that the superior *dialogue is chaining*.

This occurs in the "Completing commitment" procedure (see 11.5.3).

66) A leaf TPPM that is in the DECIDED (commit) state ceases to be part of the transaction tree when the TPPM issues a TP-COMMIT-COMPLETE indication provided that the superior *dialogue is not chaining*.

This occurs in the "Completing commitment" procedure (see 11.5.3).

67) A leaf TPPM makes the transition from the DECIDED (commit) state to the DECIDED (rollback) state when the TPPM issues a TP-COMMIT-COMPLETE indication followed by a TP-ROLLBACK indication.

This occurs in the "Completing commitment" procedure (see 11.5.3).

68) A leaf TPPM makes the transition from the DECIDED (rollback) state to the ACTIVE state when the TPPM issues a TP-ROLLBACK-COMPLETE indication provided that the superior *dialogue is chaining*.

This occurs in the "Initiating transaction after rollback" procedure (see 11.5.11).

69) A leaf TPPM that is in the DECIDED (rollback) state ceases to be part of the transaction tree when the TPPM issues a TP-ROLLBACK-COMPLETE indication provided that the superior *dialogue is not chaining*.

This occurs in the "Initiating transaction after rollback" procedure (see 11.5.11).

A leaf TPPM that is in the DECIDED (rollback) state also ceases to be part of the transaction when the TPPM receives a TP-BEGIN-DIALOGUE (rejected) response.

This occurs in the "TP-BEGIN-DIALOGUE response" procedure (see 11.3.3).

70) A leaf TPPM makes the transition from the DECIDED (commit-one-phase) state [or DECIDED (unknown) state] to the ACTIVE state when the TPPM issues a TP-COMMIT-COMPLETE ind (or TP-UNKNOWN-COMPLETE indication, respectively) provided that the superior *dialogue is chaining*.

This occurs in the "Completing one-phase and read-only" procedure (see 11.5.5).

71) A leaf TPPM that is in the DECIDED (commit) state [or DECIDED (unknown) state] ceases to be part of the transaction tree when the TPPM issues a TP-COMMIT-COMPLETE indication (or TP-UNKNOWN-COMPLETE indication, respectively) provided that the superior *dialogue is not chaining*.

This occurs in the "Completing one-phase and read-only" procedure (see 11.5.5).

Annex G

Managing association pools by inference

G.1 Introduction

The performance of a TP system is directly affected by how fast a dialogue can obtain an association. To expedite this process, TP maintains pools of associations with other TP systems. There are two ways to coordinate the building and maintenance of these association pools:

1) through the exchange of protocols between the two TP systems, or

2) through the observed behaviour of the two TP systems.

The following mechanism coordinates association pools by monitoring the behaviour of the peer TP system as associations are opened and closed. This method has the following benefits:

a) easy and inexpensive to implement;

b) yields similar results to protocol-driven methods;

c) inter-operates with peer TP systems which have NOT implemented the same mechanism; and

d) can be manipulated by systems management mechanisms.

The mechanism specified in this annex is given as an example of management by inference. Implementations are free to use this, or another scheme of greater or lesser complexity.

G.2 Definitions

The mechanism described below operates in groups of associations which share common properties. It is an implementation choice as to which properties are common to the associations in a single group. According to such a choice, the associations in a group could be:

a) all the associations, to any peer AE;

b) associations with a particular peer AE;

c) associations to a particular peer AE using the same application context;

d) any to the above, constrained to have the same contention polarity (i.e. all have this AE as contention-winner or all have this AE as contention-loser).

NOTE 1 – The associations in a group include both those that are free and those that are in use.

NOTE 2 – The term "pool" can be used to describe any of these groupings, or appropriate combinations. It can also be used to refer to the associations that are currently free.

For the contention-winner/loser distinction, an implementation can choose to consider only the associations on which it is contention-winner, or can manage both the contention-winner and contention-loser associations. In this latter case, since this side can establish a dialogue over either a contention-winner or a contention-loser association (possibly after bidding), the free associations in the winner and loser groups can be considered parts of a single pool. When selecting an association, the APM could choose a contention-winner association if one is available, dropping to a contention-loser is necessary, but only create a new (winner) association if neither are available.

NOTE 3 – Implementations can treat the two winner and loser groups as a pair for configuration purposes.

The Association Pool Manager (APM) uses the following information for each association group:

in-use: The count of in-use associations

free: The count of free associations

min: The minimum number of associations to maintain in the group

norm: The normal number of associations to maintain in the group during normal demand

max: The maximum number of associations the group is permitted to contain

Where min <= norm <= max.

Within the group, the APM distinguishes "fixed" and "variable" associations. Fixed associations are not distinguishable from variable associations by the peer except by their long-lived nature.

The APM attempts to support at all times a number of associations (called "fixed" associations) up to the number specified by "norm". These associations may or may not exist at any point in time, but the APM will accept their establishment whenever they are needed, and the APM does not release them when associations are required to other AEs.

The APM allows additional associations (called "variable" associations) to be established when all of the fixed associations are busy, but the APM can release these (after a specified period of time) if associations are required to other AEs.

- *Establish an association* – To issue an A-Associate request and wait for an A-Associate confirm.
- *Release an association* – To issue an A-Release request and wait for an A-Release confirm.
- *Allocate an association* – To use an association that has already been established and is currently *free*.
- *Free an association* – To stop using an association and make it available for future allocation.

G.3 Rules

a) The implementation can support either fixed associations, variable associations or a combination of both.

b) For each group the following values determine the fixed and variable portions:

   – min: Minimum fixed associations, where min >= 0, represents the lowest number of "fixed" associations.

   – norm: Maximum fixed associations, where norm >= min, represents the largest number of fixed associations.

   – max: Maximum number of associations, where max >= norm, represents the maximum number of associations a group can contain in total (fixed + variable). This value may change over time depending on availability of resources and the responses of the peer system, so the effective max value at any point in time is referred to as max_t in the following text.

   These values may be configured locally, or manipulated by systems management.

c) For each group the following timers prevent thrashing between the AEs:

   – T-norm: The length of time an association must remain unused before it may be *released*, if the total count of associations in the group is > min and <= norm.

   – T-max: The length of time an association must remain unused before it may be *released*, if the total count of associations in the group is > norm.

   Where T-max <= T-norm.

   – T-est: The length of time to wait before retrying association *establishment* after failure to establish an association.

   These values may be configured locally, or manipulated by systems management.

d) Fixed associations are pre-configured for each group and the resources to support these associations are always available. Fixed associations are *established* before variable associations.

e) Variable associations are *established* as needed (when all fixed associations are busy) if the necessary resources are available or if they can be made available by *releasing* another variable association.

f) The Association Pool Manager allocates available associations for use in order of most recent usage. That is, most recently used first.

g) When an association is freed and the count of established associations (free and allocated) in the group is > min the re-use timer is started for that association. If the number of associations in the group is > norm, then the re-use timer uses the value T-max. If the number of associations in the group is <= norm and > min, then the re-use timer uses the value T-norm.

h) Associations are never released if the number of established associations is <= min for the group. Otherwise, associations are released when the re-use timer has expired for the association.

i) It is assumed local resources for associations are allocated dynamically from a common set for all groups. If resources are not available for an association it will not be established. This is true for both the initiating and the responding AE. In the event that a high priority association is required, a lower-priority association which is actively involved in a dialogue branch may be aborted, based on a local decision to do so.

j) Associations are released by A-Release. A-Release indications are always accepted.

k) If resources are not available to support an incoming Association indication (indicates that max_t is exceeded), then a negative Association response (diagnostic=no_reason_given) is issued.

l) If a negative Association confirmation (diagnostic=no_reason_given) is received, this indicates that the peers "max_t" level has been exceeded, so no additional Association requests should be issued for a locally defined period of time (T-est), to prevent thrashing. It is a local issue whether a request to allocate an association fails or waits for an association to become available while the T-est timer is running.

G.4 Benefits

TP Protocol Machines, specifically Association Pool Managers, which follow the above rules will minimize thrashing of association establishment requests caused by differences in the sizes of paired association groups (i.e. corresponding groups at the two AEs). This permits groups to be locally configured with minimal coordination. The result is that the minimum configuration of two paired pools is adopted with room allowed for temporary expansion of the pool when needed.

The various timers which are described prevent the Association Pool Managers from releasing associations which have a high rate of use, and from attempting to establish low-priority associations when higher-priority associations are needed by the peer. The re-use timer ensures that associations have at least a short life-span which can be extended indefinitely by a sufficiently high rate of use. The T-est timer backoff mechanism stops an AE from persistently establishing unwanted associations.

Implementations which do not follow these guidelines for association pools will inter-operate with implementations that do; however, the local configuration of pools must be closely coordinated. If not, significant network overhead may be introduced by persistent attempts to establish associations which cannot be supported.

G.5 Suggested system management objects

G.5.1 Meters

- Number of outbound Associations successfully established.
- Number of incoming Associations successfully established.
- Number of outbound Associations which failed.
- Number of incoming Associations which failed.
- Number of Associations released by peer.
- Number of Associations released locally.
- Number of Dialogue requests which failed locally because an association could not be established.

G.5.2 Alarms

- The number of association release requests and association establishment indications in a given time exceeds the threshold. (The value of "norm" and/or "max" for the pool may need to be increased.)
- The number of association release indications and association establishment requests in a given time exceeds the threshold. (The peer's value of "norm" and/or "max" for the pool may need to be increased.)
- The number of association indications which are refused in a given time exceeds the threshold. (The value of "norm" and/or "max" should be increased for the pool.)
- The number of association requests which are refused in a given time exceeds the threshold. (The peer's value of "norm" and/or "max" should be increased for the pool.)